TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

SYED FAROOQ HASSAN

MODELING INFRASTRUCTURE MAINTENANCE CONTRACTS IN A GEOSPATIAL DATABASE

Master of Science Thesis

Examiner: Professor Kari Systä
Examiner and topic approved on 1 November 2017

ABSTRACT

SYED FAROOQ HASSAN
Tampere University of technology
Master of Science Thesis, 62 pages
December 2017
Master's Degree Program in Pervasive Computing
Major: Software Engineering
Examiner: Professor Kari Systä

Keywords: Road maintenance, DigiRoad, geographic information system, data matching, data fusion, digitalization

Automation in every field has introduced various means to increase productivity and efficiency. Recently, the road maintenance industry has adopted automation. However, automated scheduling, pricing, budgeting, reporting and invoicing of work are restricted by the provided format of data. In addition, the available information to depict real state of environment and capabilities of machines is also one of the obstacles to automate the aforementioned processes. This research work addresses these problems by modeling the data in database that will automate the processes.

This thesis focuses on studying existing models of road. The information from existing road models is extracted and compared with each other. Afterwards, the necessary data comprising of work requirements along with related machinery is studied, and the useful information is extracted. The road model is then fused with the machine and contract data model. This fusion results to give information of roads, machines and work requirements which are missing in currently available information. The implemented approach is validated with the help of real world contract area. This results in providing the model which will automate the maintenance processes. The resulted model is aligned with the expert view.

PREFACE

This research work in this thesis has been carried out in the company Nosteco Oy.

I am really thankful to Kari Systä for his guidance throughout my thesis work. I also would like to thank Jari Nikara, Janne Pietiäinen and Kimmo Kuusillinna for introducing me to this field which was unknown to me before. It is due to their patience and competence that I have learned and being able to complete my work in the given time. Moreover, a lot of credit also goes to my friends who have helped me by offering advice and support.

In the end, I would like to thank my wonderful family. Without my parents love, support and prayers this would not have been possible.

Syed Farooq Hassan

28th December 2017.

Tampere, Finland

CONTENTS

List of Figures

List of Symbols and abbreviations

CSV	COMMA SEPARATED VALUES
DBMS	DATABASE MANAGEMENT SYSTEM
ETRS	EUROPEAN TERRESTRIAL REFERENCE SYSTEM
FTA	FINNISH TRANSPORT AGENCY
GDAL	GEOSPATIAL DATA ABSTRACTION LIBRARY
GIS	GEOGRAPHIC INFORMATION SYSTEM
JPEG	JOINT PHOTOGRAPHIC EXPERTS GROUP
JSON	JAVASCRIPT OBJECT NOTATION
KDD	KNOWLEDGE DISCOVERY IN DATABASE
MBR	MINIMUM BOUNDING RECTANGLE
OGC	OPEN GEOSPATIAL CONSORTIUM
PDF	PORTABLE DOCUMENT FORMAT
PNG	PORTABLE NETWORK GRAPHICS
QGIS	QUANTUM GIS
SQL	STRUCTURE QUERY LANGUAGE
WGS	WORLD GEODETIC SYSTEM

List of Specific Terms

DIGICONTRACT	A database table containing the work requirements in digitalized form.
DIGIMACHINE	A database table containing the capabilities of real maintenance machines in digitalized form.
DEFINITION OF DONE	A database table that evaluates the completed work in a region using the digitalized machine data, maps and work requirements.
DATA FUSION	The process of combining data sources to produce more accurate results.
FEATURES	A representation of real world objects on map such as bus stops, barriers.
GEOMETRY	The point object, linestring object, polygons object that represent spatial data are commonly referred to geometry.
LINESTRING	The linestring object is formed by connecting the line between the point objects.

1. INTRODUCTION

This chapter presents the motivation and background of this thesis. It illustrates the identified problems. The chapter also presents the concrete objectives. Moreover, it explains the implemented proof of concept and the results. The last part of this section presents the structure of thesis.

1.1 Motivation and Background

Majority of road maintenance is done as contracts where contractors provide maintenance work as a service for an ordering authority. For instance, the Finnish roads are divided into 80 contract areas that are maintained by contractors [1]. The contract is described as a set of documents: maps, work, safety and environmental requirements. This set of documents is also called contract data. The contract data is available in human readable format and therefore, require manual review and processing. The manual review of these instructions and requirements may lead contractors to wrong estimations of resources, budget and pricing.

The contract data is rarely available in machine readable format, and thus limiting exploitation of automation. The amount of contract data is huge and comes from several sources such as from numerous ordering authority databases and file systems. This makes effort estimation and follow up of work progress challenging and inefficient. To achieve automation of processes, the contract data needs to convert to a digitalized form that computers can easily access. The technique of digitalization has been used to achieve the aforementioned goal. The digitalization allows the contractors to autonomously estimate resources, budget, pricing and later on track the work progress.

Lately, a few projects and papers have discussed the challenges associated with the automation of road maintenance processes because of available contract data format. According to [3], an estimated 80 percent of business to business transactions are unpinned by contracts. Managing the contracts is important for the success of business [3]. In [13], a poll has been conducted which indicated that most of the organizations use combination of hard copy files and documents to manage contracts [13]. Reviewing and processing the hard copy files and documents include risks and chaos such as inability to meet contract requirements, delayed financial decisions and other penalties [12]. Moreover, the processing of human readable files and documents also constitute a significant cost to organizations [14]. Therefore, organization struggles with monitoring the timeliness, requirements, terms and conditions in contracts [12]. To address these aforementioned problems the contract data should be digitalized.

1.2 Problem Description

Contractors spend a lot of time to estimate resources and to verify that work has been completed according to the work requirements. Contractors are forced to use human readable format description when estimating cost, checking requirements, and identifying the status of work progress. However, human readable format requirements and specifications make the problem worse than by providing the bunch of tables and files. It also requires extreme concentration but still estimates can be inaccurate.

To create this automated estimation and tracking, the main problems that need to be tackled are given below:

- Contract data is not digitalized to automate the estimation of resources and tracking of work progress
- Existing data sources describes the maintenance work partially and for having a complete view requires combining several data sources and managing their relations.
- The automated estimation and tracking of work progress requires the road features and machine data. The road features show the real state of environment whereas, machine data shows the capabilities of machines. However, the current tracking systems do not exploit or show the relations of road features and machine data.

1.3 Objectives

The main objective of this thesis is to enable automated estimation and tracking of maintenance work progress. To meet this objective we will focus on following:

- Study existing road and street database (DigiRoad) and digitalize the contract data and machine data
- Fusing road data, machine data and contract information to form a data description to describe the maintenance work and its environment.
- Defining the definitions of done for maintenance work that allows later to track the work progress and to ensure requirements are fulfilled.
- Implementing the proof of concept prototypes to demonstrate the approach.

1.4 Structure of Thesis

The remaining thesis is divided into five chapters that are structured as follows:

- Chapter 2 enlightens the reader about the techniques and concepts that has been used in this scope of study.
- Chapter 3 elucidates the methodology that has been used to accomplish the set of goals.
- Chapter 4 benchmarks the implementation techniques and technologies used in the thesis to compose a model for road maintenance.
- Chapter 5 provides the results that are obtained as a result of implementation techniques.
- Chapter 6 explains the conclusion deduced from the results and the future work needed on this study.

2. BACKGROUND KNOWLEDGE

This chapter presents the background to carry out the research work. In this chapter Geographic information system (GIS) is explained in detail. Moreover, this chapter highlights the role of databases, data fusion, spatial data matching, data cleansing and data analysis.

2.1 Geographic Information System

Researchers have provided different definitions for the term GIS [4]. In this thesis, the following interpretation of GIS is used:

‘‘A GIS is a computer based information system that provides tools to collect, integrate, manage, analyze, model, and display data that is referenced to an accurate cartographic representation of objects in space.” [7]

Recent trends show that GIS has been used extensively in the business domain because it is a powerful tool for exploring information that has been locked in the data [2]. The explored data contains the location information [5]. GIS analyzes this data as input and produces the presentations from the given data. In addition, this data can be presented in many coordinate systems as GIS supports a large set of coordinate systems and ways of presenting geo referenced data [6].

GIS represents the real world objects such as lakes, rivers and forests in the digital format [17]. These representations are formed by the data which is stored in computer memory in the form of bits and bytes, known as spatial data.

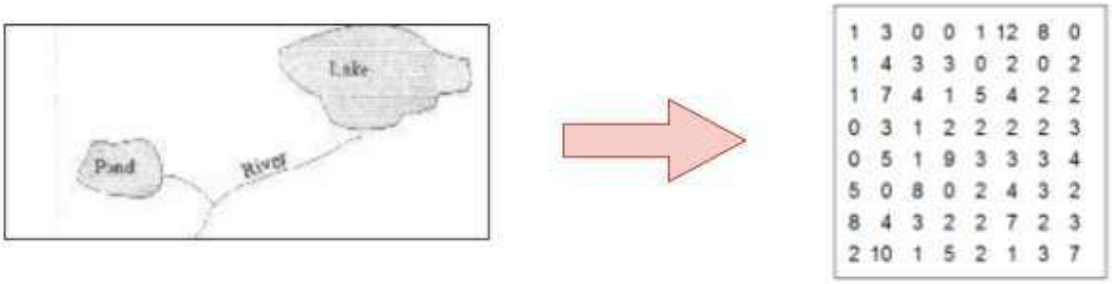
2.1.1 Spatial Data

Spatial data is defined as the data that identifies the location of different features on earth. The objects such as rivers, lakes, cities, countries, roads and so on are termed as features on earth. The spatial data is formed by combination of points, lines, rectangles, surfaces, and data of higher extension including time [11]. Two models of spatial data exist that affect the volume and speed of the processing data, which are: Raster data and Vector data.

Raster data is overlaid on the grid that covers the region of real world [15]. The raster dataset partitions the map data in the form of grid. The grid contains different cells whereas, each cell shows different value of map data that is depicted in Figure 1 [16]. In

Figure 1, the map data is converted into a grid containing different cells. The amount of data stored in each cell depends upon the size of each cell [15]. Each single cell represents the size of area which is called raster resolution.

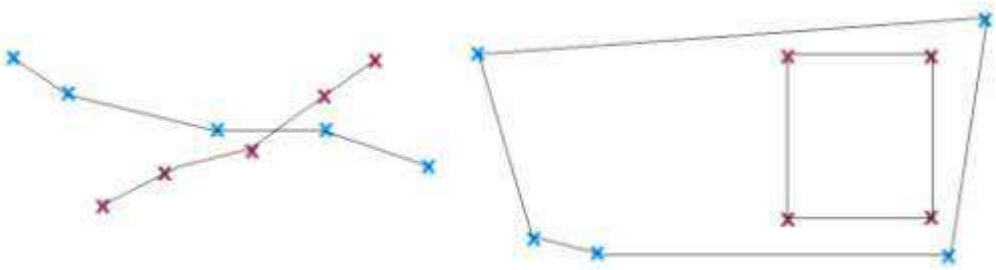
The scanning maps, digital images and remote sensing images can easily be acquired using raster data [16]. The raster model is appropriate for data which varies considerably. In addition, raster data type is only suitable for application when continuous space is used such as temperature, rainfall, soil type or elevation [16].



(a) real map data ***(b) raster data***

Figure 1: Conversion of real map data into raster data [16]

Vector data contains the location of the point which is stored as coordinate pairs (x,y) [17]. Each point is referenced by the appropriate coordinate system [15]. In contrast to raster data type, vector type stores only the interesting feature therefore, the map size in vector data is smaller. Vector data can be classified into three types [16]: points, arcs and polygons



(a) Arcs ***(b) Polygon***

Figure 2: Structure of vector data

The fundamental primitive in vector data are points [16]. In addition, arcs are formed by combining different points whereas, structure of polygon is defined by set of arcs. In Figure 2(a), a complete arc can be seen which is formed by joining several points. Similarly, set of arcs are joined to form a polygon in Figure 2(b). Different crosses in Figure 2(a) and Figure 2(b) represents the points in the vector data. This thesis only focuses on vector data type as raster data type is beyond the scope of this research work.

Vector data uses geometrical objects to represent the shape of real world map features. These geometrical objects are also known as spatial objects. These objects either exist as single entities or are formed by the combination of other objects [15]. Similarly, these can either be in two-dimensional or three-dimensional spaces. In addition, spatial objects are classified into three types: point object, linestring object and polygon object. The point object, linestring object, polygons object that represent spatial data are commonly referred to geometry.

Point is the simplest geometric object that is used to represent different features of real world map such as bus stops, houses, traffic lights, pedestrian crossing, barriers and railway crossings [15][16]. The features including bus stops are shown in Figure 3(a). In Figure 3(b), the features like bus stops are converted to vector data objects such as point object.

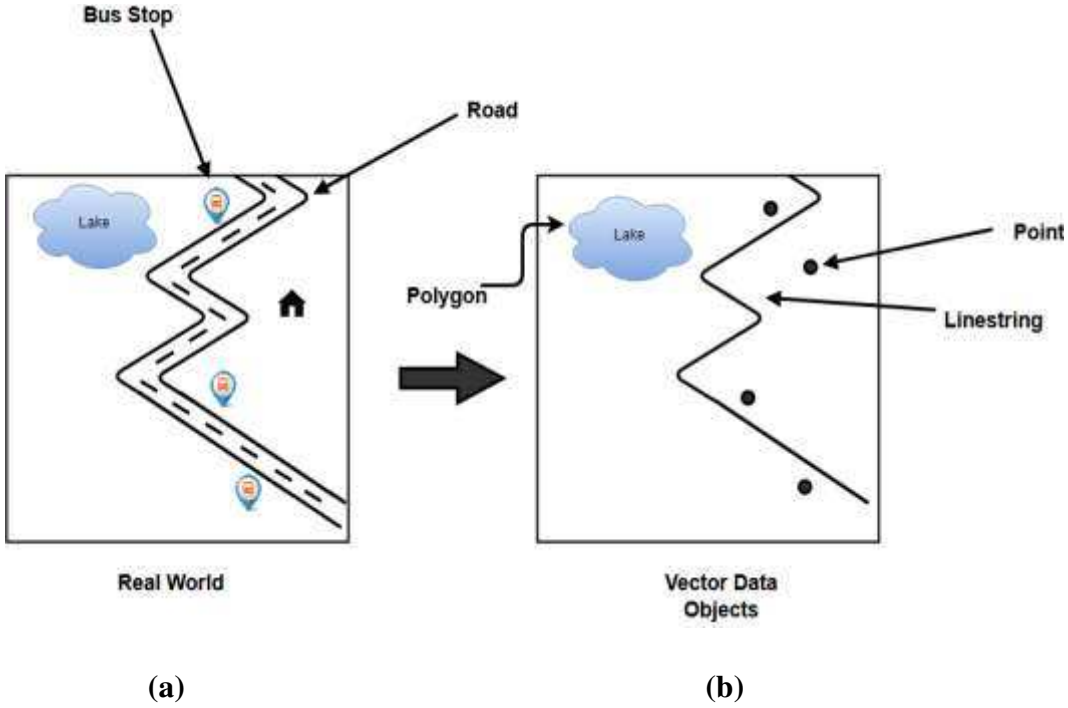


Figure 3: Vector Data Objects

The point data contains the x and y coordinates of the real map enclosed in a point object. The points attribute data can be converted into different geometrical objects depending upon the need of the application.

The linestring object is formed by connecting the line between the point objects. The linestring is considered as one dimensional extension of points [15]. Each vertex of linestring contains the x and y coordinates of real map. In addition, multiple lines can be combined to form a list of linestring also known as multilinestring. Linestrings are used to represent the linear shape of map objects such as roads, streets, rivers and so on [16]. In Figure 3(a), a road is shown which is represented as linestring in Figure 3(b).

A polygon is a bounded shape that covers extended region and whose boundaries contains homogenous area. Polygon is used to represents different features on map such as lakes, states and villages [15]. In Figure 3(a), a lake is converted to the bounded shape in Figure 3(b). Similar to linestring, different polygons can be combined to build a multipolygon.

2.1.2 GIS Data Representation Formats

GIS data can be represented with help of different formats. These formats include Cartesian coordinate system, shapefile, GeoJSON, keyhole markup language (KML) and many others. However, shapefile formats [18] and GeoJSON format [67] are only considered in this research work.

Shapefile is a format of storing the spatial objects and their respective attributes [18]. Shapefile is formed by the combination of graphic format files, graphic index files, attribute data format files and projection files [19]. Shapefile has the support to store object such as linestring, point and polygon [18]. The main advantage of shapefile format is that it is commonly used format in GIS as well as it is also compatible with most of the software's [20].

On the other hand, GeoJSON format uses JavaScript object notation (JSON) to represent spatial objects along with their attribute information. JSON is defined as the light weight data interchanging format. These spatial objects include point, linestring and polygon [21]. One of the reason to use GeoJSON format was to distribute the data over the web [21].

2.1.3 Geographical Coordinate System

A geographical coordinate system is a spatial reference system that uses three dimensional spheroid surfaces to place the geometry. Each point in three dimensional spheroid surfaces contains the longitude and latitude. Therefore, it forms a complete coordinate system. [15].

There is a need to understand the coordinate system in order to identify the position of spatial objects on map [22]. To represent the position of different objects on map, different coordinate systems are used. The coordinate system used globally is known as World Geodetic System (WGS) [68]. It is expressed in three-dimensional form as Cartesian coordinates [22]. In contrary, European Terrestrial Reference System (ETRS) [69] is utilized primarily for European areas. GIS uses these coordinate systems to utilize the position of geometries from the map. Similarly, there are several other coordinate systems that can be used depending upon the need of the applications.

2.2 Databases

Database is defined as [30]:

“The collection of data that is organized in such a way that it can be easily managed, accessed, updated and retrieved”

Storing large volume of persistent data forms the major characteristic of database. This means that this data can withstand the unexpected software and hardware failures except for few disk failures [30]. Databases has been playing the role of warehouse for different businesses and accounting data [25]. Information related to employees, customers and products can be easily stored in the databases [25]. The data stored in these databases can be simple numbers, names and addresses, etc. [25].

The databases can be seen as a memory device such as disk containing different files. Certainly, it would be possible to access these files and write directly to these files. However, this could cause problems pertaining to security, uncertainty, concurrency and complexity of data manipulation. Therefore, to access these files a collection of software has been created known as Database Management System (DBMS). The tasks performed by DBMS are: [30]

- Defining the database
- Database construction
- Manipulation of database
- Retrieving specific data using queries
- Performing updates on database

Figure 4 illustrates the environment of the DBMS. It shows how DBMS acts as a bridge between the user and application program [30]. The DBMS software contains two parts. The first part processes the query and the other part accesses the data and metadata necessary to understand the structure of database [30].

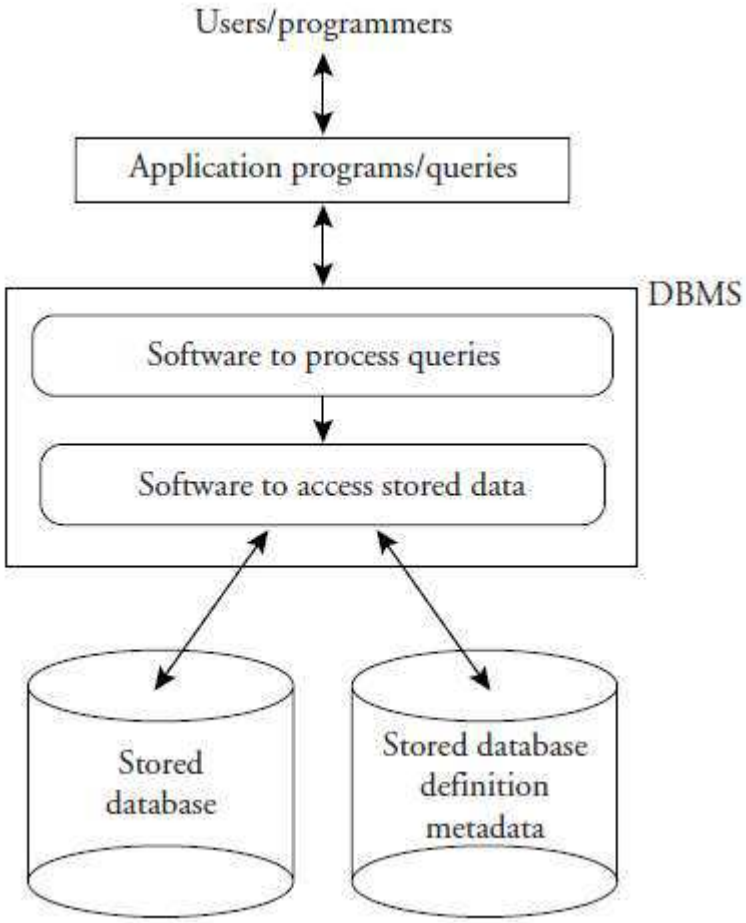


Figure 4: Environment of database system [30]

Database schema shows how data should be organized in the database. Moreover, it defines the skeleton structure which represents the logical view of entire database. Once the database schema is defined, data in database can be easily inserted, retrieved and deleted by a set of commands known as queries [30].

The operations of DBMS are dependent on the type of model used. Currently, the most commonly used model is relational model. The relational model depends upon the structure known as table or relation [30]. Therefore, using DBMS the conventional databases has provided an efficient way to store and retrieve the non spatial information.

2.2.1 Spatial Databases

The processing of spatial data is becoming increasingly dependent on DBMS. The use of DBMS has not only changed the fundamental concepts of spatial data and management but it has also made the way for new skills and demands of spatial data managers and users [31]. The major features of using DBMS in GIS are [31]:

- The spatial data can be represented in the form of tables. A spatial object is always included in each row of the table. The attributes are given in columns.
- Attributes of spatial data can have alphanumeric types such as integer, character, boolean, and so on.
- The querying of spatial data is based on the structured query language (SQL) language.

The spatial databases are defined as the databases that are optimized to store, manage, update, and retrieve spatial objects. These databases are capable of doing all query tasks and data modeling as available in conventional databases. Spatial databases are considered as an extension of the ordinary database system. In addition, spatial databases provide some extra features as compared to conventional databases. These features are: [31]

- It can store the spatial objects such as polygon, linestring and points.
- It processes the complex data types by using spatial operators that are more sophisticated.

The management, collection and use of spatial information are not possible without the power of spatial databases. However, spatial databases store large volume of spatial information, this stored spatial information is computation expensive. Processing large amount of spatial information has not yielded the best results. Therefore, spatial indexing technique is used to make processing efficient. [31]

Spatial indexing is used in spatial databases to optimize the process of accessing and returning the data to users. Due to large quantity of spatial data in databases, the computations such as select or join are highly inefficient [32]. For example, a spatial dataset might contain thousands of point and linestring objects [32]. Moreover, spatial data can also have complex structure and relationships [32]. The complex structures and relationships are formed when thousands of spatial objects are adjacent or intersecting with each other [32]. Therefore, the computation speed of queries to manipulate complex spatial relationship in spatial database relies upon the process of indexing [32].

The concept of spatial indexing is that it gradually narrows the area of spatial access until the objects in the database are found [31]. Due to large volume of data in a spatial dataset it is highly inefficient to iterate through all the dataset and find the spatial relationship among the data [32]. In order to find spatial objects in an efficient manner, it is essential to have index over location [32].

Numerous methods have been used for spatial indexing in spatial databases. However, one of the commonly used method is R tree algorithm. The hierarchy of R tree algorithm is shown in the Figure 5. In this algorithm, multiple rectangles are stored at each node of multi level tree, which can be seen in Figure 5. The rectangles are stored in the

root pages, branch pages and leaf pages of the tree. The root pages are present on the top whereas, branch and leaf pages are present in middle and at the end of the tree. The minimum bounding rectangles (MBR) are formed across the each individual data objects. As shown in the Figure 6, the object A, B and C are bounded inside the rectangles R9, R10, R11. The index file in R tree algorithm stores the reference to the MBR and coordinates of MBR corners. [31]

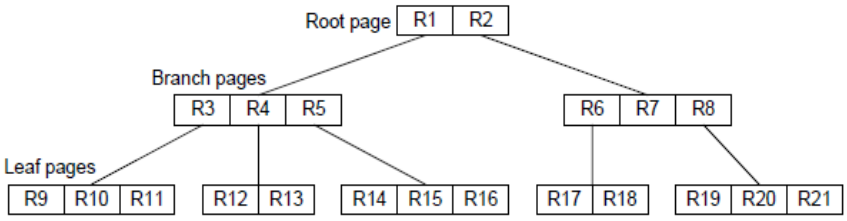


Figure 5: Hierarchy of R tree indexing [31]

If the database receives the request for spatial search, there will be search within the root pages consisting of coordinates. The scanning will determine that which MBR at branch level falls within window. Once the identification of MBR at branch level is completed, the next scan identifies the MBR object at leaf level. The MBR object at leaf level is identified by search through rectangles. This scan will identify MBR falling within the boundary of search window. The system will then use the reference information between MBR and identification number of object to give access to tables that contains all the information. [31]

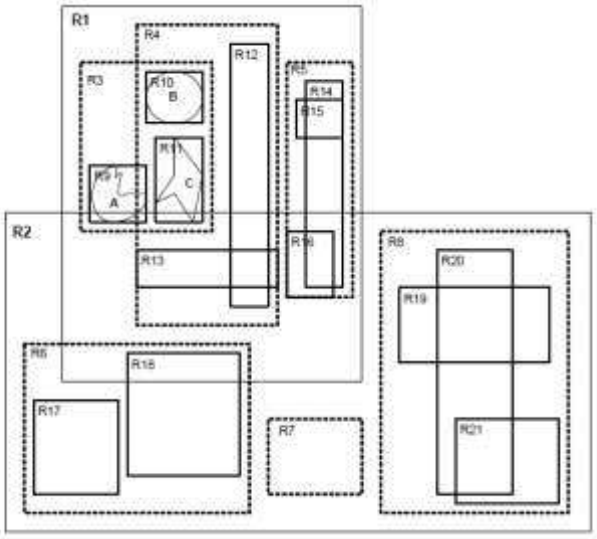


Figure 6: Bounding boxes spatial relationship [31]

2.2.2 POSTGIS

PostgreSQL [70] is a traditional relational database. To store spatial data in PostgreSQL an extension named as PostGIS was created [26]. PostGIS offer different spatial functions such as area, intersection, union to manipulate the spatial data. PostGIS is similar in functionality to many other spatial databases such as SQL Server Spatial support, Oracle Spatial. However, PostGIS provides better performance as compared to these databases. PostGIS support all spatial objects such as linestring, polygon, points, multilinestring, multipolygons, and so on.

2.2.3 SpatiaLite

SpatiaLite is an extension of SQLite for the purpose of storing, processing and retrieving spatial data. SQLite is a light weight personal DBMS and SpatiaLite is built on top of it. The whole SQL engine can be embedded in the application by using SQLite and SpatiaLite. SpatiaLite contains only a single file, which can be easily transferred from one system to another. In addition, the operating system of origin and destination system can also be different because SpatiaLite also supports cross platform portability. SpatiaLite functions are quite similar to PostGIS therefore, switching between them is not a problem. SpatiaLite also supports manipulation of all spatial objects such as linestring, polygon, and point. [33]

2.3 Data Fusion

The sensors used in different industries have generated a lot of data. This data need to be processed in a meaningful way. Due to the increase in demand of accurate information, a robust technique is needed to manage the data efficiently. Therefore, a technique known as data fusion has been used to integrate the information from various sources. [61]

Recently, data fusion has started to become popular in various fields. The domain of data fusion is large therefore, it cannot be restricted to a single field. However, it's very challenging to provide a precise definition for data fusion but most commonly it is defined as the following [58]:

"A process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats, and their significance. The process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results"

Data fusion aim is to produce accurate and high quality results by integrating the data from various sources [59]. The concept of data fusion is shown in Figure 7. It combines different data sources s1, s2 and s3 to produce a combined data source called fusion system. The result deduced from this source is called fusion result.

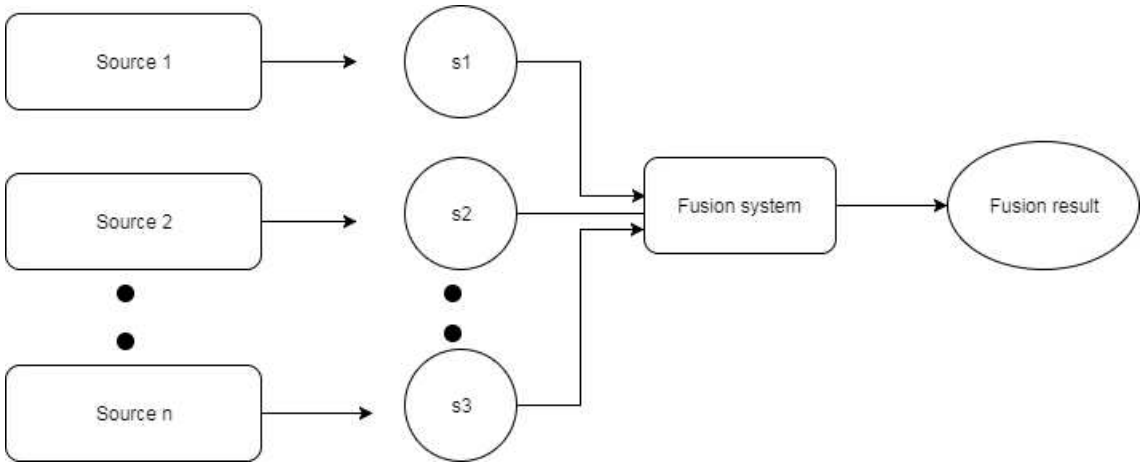


Figure 7: Data fusion

Due to different types of inputs data fusion is divided into various types in [64]:

- Complementary fusion
- Redundant fusion
- Cooperative fusion

2.3.1 Complementary Fusion

In complementary fusion, the sources which do not depend on each other are combined in order to give complete image of phenomenon under observation. In the Figure 8, it can be seen that independent sources S1 and S2 are providing the different data “a” and “b”. The piece of data from both sources are then fused together to give more accurate and broader results as “(a+b)”. [65]

2.3.2 Redundant Fusion

In the redundant fusion, the data from two or more sources providing the same information can be fused together to give more improved results. Redundant fusion increases the accuracy, confidence and reliability of the given data. In addition, redundant fusion also improves the quality of data. In the Figure 8, it can be seen that sources “S2” and “S3” provides the same data and the data is fused together to get more accurate data “(b)”. [65]

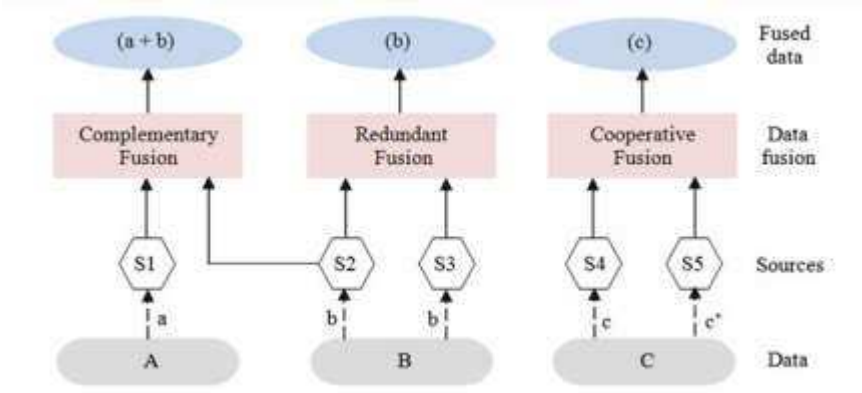


Figure 8: Types of data fusion

2.3.3 Cooperative Fusion

The data fusion is known to be cooperative when independent sources are combined to give information which would not be available from single source. In Figure 8, it can be seen that sources "S4" and "S5" provide different results as c and c*. These results are fused together to provide the result as "c" which describes the scene better. Cooperative data fusion technique should be carefully applied because the resultant data might have different inaccuracies and errors from all the combining data sources. [65]

2.4 Spatial Data Matching

The process of bringing together different data sources that show similarity between the data is known as data matching [38]. The techniques of data matching also help to evaluate the quality of data. The need of matching has been increased due to the usage of different data sources in different application domains [39]. Recently, data matching has started to play an important role for the location-based applications on the basis of spatial data.

Accurate data is a basic necessity for meaningful data analysis. In geographical information system, it's rare that spatial dataset representing real world environment is similar to the actual real world environment [42]. Therefore, spatial data analysts always collect data from different sources to provide a meaningful representation of data.

To provide the complete view of spatial data, the data is collected from different sources. This results in storing of data in different formats in the database. Therefore, these data sets are termed as highly inconsistent. For example, the same road data stored in different databases can be of different format leading to inconsistencies [40]. Therefore, to create a single view from these highly inconsistent data sets there is need to use the technique of matching.

Spatial data matching is defined as [36]:

“The process of identifying, matching and merging corresponding spatial features to same entities from different spatial dataset”.

Data matching is considered as an essential step in the process of spatial data comparison [36]. Spatial datasets are quite often huge and matching often becomes difficult. The spatial datasets not only require accurate matching between the datasets but also requires the completion of task in minimal time. With usage of diverse spatial databases, the geo data matching has become complex.

2.4.1 Methods of Matching

There are different methods of spatial data matching. Each method of matching has its own importance and can be used in different conditions depending upon the spatial dataset available.

The most popular method of matching between two objects in a different dataset is Euclidean distance method or Hausdorff distance method. The Hausdorff distance calculates the maximum distance between each point of one linestring with the point of other linestring in a dataset. Linestrings are regarded as a corresponding pair if the distance value is less than the threshold value. [42]

In [44], an area based method was proposed for matching to remove the differences between datasets. Each dataset contain an area represented by spatial data. The area which has distance less then threshold is considered as potential matching pair. In [44], the cost function was created to evaluate the potential matching pairs. The cost function was calculated using different parameters such as size of area, centre of gravity, and number of linestring segments. The evaluation with cost function forms a list of ambiguous matching pairs. The list is sorted by the costs with the lower cost on top of list. The matching pair with the lowest cost is used for final results while eliminating the other matching pairs containing higher cost. [40]

The point based method of matching considers the close points that require matching. In this algorithm, road network is represented as linestring and code is assigned at each intersection of linestring. The code consists of street name, the coordinates of points or the number of linked edges. Each intersection is assigned with similar code in the dataset. The assigned codes are then compared within different datasets. The intersections within these datasets are also assigned with the most similar codes. In this way, references between the datasets can be derived.

For point dataset a location based join algorithm has been developed in [43] for matching. It is capable of matching two or more dataset at the same time either sequentially or

simultaneously. Different performance of algorithm is presented based on recall and precision.

Buffer intersection matching involves the matching of two datasets which are geographically close to each other [41]. In most cases, it involves the matching of datasets which are represented as linestrings. In Figure 9, an example of buffer intersection method is shown. The roads from first dataset are represented as linestrings in Figure 9(a). In this method, the linestrings in the first geometry are extended by using a bounded region known as buffer. In Figure 9(b), similar linestrings in Figure 9(a) are extended to create a buffer around the geometry. In Figure 9(c), red line represents the region of second dataset that is placed over the buffer region of first dataset. In Figure 9(c), it can be seen that some of the red linestring of second dataset are not completely enclosed in the buffer of first geometry. The intersection analysis is then applied between the first dataset represented as buffer and second dataset represented as red linestrings. This produces a merged geometry of two areas. This method is very useful in matching the areas however, it produces some new roads as a result of matching. This can be seen in Figure 9(d). The buffer size plays an important role in this method because a larger buffer might add extra new roads. [41]

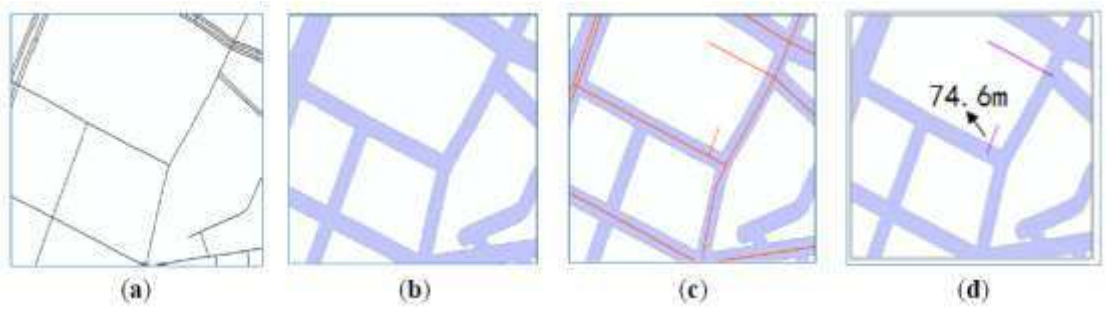


Figure 9: Buffer intersection method

The method known as nearest neighbor pairing is also used for matching. This method detects the nearest neighbor of the object regardless of the distance between them. For example the linestring B in one dataset is closer to the linestring A in other dataset. Therefore, both linestrings are considered as matched pairs. [42]

In GIS, topology specifies the relationship between spatial objects. In [45], a mixed approach was proposed which uses topology. It combines both point based and line based method to match the datasets. The node matching approach utilized the topological properties for identifying the counterpart nodes. Proposed method performs a walk from one node say A in a dataset 1 to its adjacent nodes say B or C and then to corresponding node of other dataset 2 say B' or C'. The matched pair is formed if the connection from B' and C' to target A' exists in the dataset 2. [40]

In this thesis, buffer intersection method is used because the dataset involved are represented as linestrings and placed geographically close to each other. In addition, there is

a need to create a merged geometry from two datasets therefore, for this purpose a simple buffer intersection method is suffice.

2.5 Data Cleansing

The data available in the large databases might contain data quality problems. The data should fulfill certain quality characteristics [51]. The presence of errors and inconsistencies can lead to wrong results of analysis [49]. The commonly used factors of estimating quality include accuracy, timeliness, completeness and consistency [48]. Data quality is estimated based on the scores of aforementioned factors. The assessment of scores will give rise to the necessity of the data cleansing for certain dataset [51].

Data cleansing is applied especially when data from various sources are merged together. In those sources, data referring to same entity is represented by different formats. While merging sources, a source containing dirty data can also affect other sources. This problem is known as merge problem and is removed by using data cleaning. [46]

Data cleansing also known as data cleaning is a process of removing the inconsistencies and errors from the data to increase the quality of data. Data cleansing should satisfy certain requirements. Firstly, it should remove all the errors from the data and secondly, it should remove the inconsistencies and extra data included in the dataset. [46]

2.5.1 Data Cleansing Process

The data cleaning approach contains set of operations. These operations are performed on the data to remove the anomalies [51]. In this study data cleaning process has been divided into two categories:

- Defining and determining errors types
- Searching and correcting errors

The errors in the data should be searched in a planned way because they can occur at any stage during the data flow [47]. The data may require several rounds of data cleaning process before making the data accurate, error free and consistent.

In defining and determining error types, anomalies are detected within data [51]. The quality of data can be improved by understanding the error types [48]. The anomalies within data are divided into four basic types [47]:

- Lack or excess of data
- Inconsistent data
- Strange patterns in data
- Unexpected analysis results in data

In searching and correcting errors, suspected data in the dataset is searched and corrected. This process needs an algorithm or an appropriate cleaning method [47]. The applied algorithm should not only reduce data cleaning time but it should also maximize the degree of automation [47]. Choosing an appropriate algorithm is a difficult task [47]. The selection depends upon the type of error in data and problem domain [48]. The challenging task is how to improve the efficiency of algorithm and include the degree of automation [48].

2.5.2 Spatial Data Cleansing

With the usage of more location based applications, amount of spatial data is increased. However, a problem in collection of spatial data is that it comes from various sources such as from different sensors, instruments and techniques. In addition, data is also collected at different time periods. Due to these problems, format of data varies. [57]

The data stored in spatial databases are represented with spatial types and spatial relationships. There are several problems hidden within the stored spatial data. For example, data can be inaccurate, inconsistent or conflicting. Due to these problems a lot of data is of no use. To remove these anomalies, cleaning is needed. [57]

Spatial data cleansing is applied on different types of data. These data types are: incomplete data, inaccurate data, repetitive data and inconsistent data. Data is incomplete because of various reasons. For example, the records were not given when database was created or laziness in input. The completeness means how each object is compiled into the dataset. The cleaning process uses different techniques such as decision trees and rough sets to make the data complete. [71]

The inaccurate spatial data measures the difference between the observed data and its true value. The inaccuracy may be qualitative and quantitative, such as outdated data not updated in a timely manner, data which is not similar to the real world objects, data obtained from inaccurate calculation, vague data with fake values. [71]

The repetitive data shows the duplicate records of same objects in one or more data sources. Duplicate data exists because of errors such as spelling mistakes. The repetitive data is more common when data is coming from several sources. The data provided by each source include the strings or identifiers which may vary in different data sources. Identifying and removing the duplicate data not only increases the storing capacity but it can also improve the computation speed. [71]

Spatial data is inconsistent when similar type of data is stored in different formats. Data can be inconsistent when acquired from several sources. The inconsistent spatial data result in the ineffective data analysis therefore, it should be eliminated with the help of data cleaning. [57]

2.6 Data Analysis

Data is defined as [28]:

"Factual information (as measurements or statistics) used as a basis for reasoning, discussion, or calculation."

By understanding the definition it can be assumed that data can be in any format such as images, characters, numbers or recordings [28]. By examining the data, the special patterns within the data can be deduced. These patterns perceive the information which enhances the knowledge. These patterns can be obtained by using data analysis.

Data analysis is defined as the process of performing different operations on data to bring order, structure and meaning to data [29]. In data analysis, data is examined to draw conclusion from extracted information.

2.6.1 Data Analysis Process

Data analysis process includes different steps from start to end [28]. It proceeds in a linear fashion [28]. The linear process of data analysis always includes the steps:

- The problem
- Data preparation
- Data exploration and visualization

The problem definition starts with the high level questions [34]. Understanding the objectives and requirements are key for successful data analysis [28]. Different decisions are made based on these questions. Decision making requires two important questions [28]

- What data should be collected for analysis
- Which types of analysis to use with the data

Different type of questions can be formulated based on different types of data [28]. The selection of one or more appropriate questions allows the process of data preparation to proceed [28].

The second step in data analysis is data preparation that includes the process of data collection [34]. Data collection is a process in which researcher collects the information to answer the research problem. Data is collected using various sources such as from documents and files [28]. In order to proceed in a right way, data should be collected in a systematic way [35]. The data collected in haphazard manner might not answer the questions related to research [35].

The data collection process is illustrated in the Figure 10. The data collection process includes different steps. The first step is named as sources which finds what type of data is needed to answer the research problem and from what sources the required data can be found. The second step is named as method which answers different questions such as who will collect the data or where data will be collected. In the third step data is collected from sources whereas in fourth step data includes the process of compilation. In the last step data is stored in databases and is ready for further analysis.

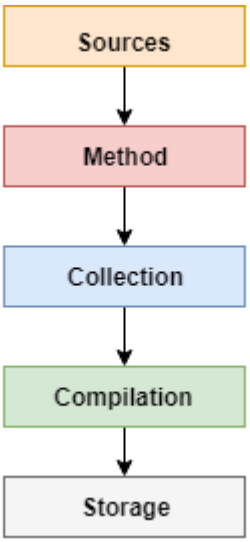


Figure 10: Data collection process [28]

In data exploration, data is explored in a statistical way to find patterns, connections and relations among the data. The data is explored by visualizing it and finding out the certain patterns within data. Visualization and exploration data holds key for decision making in organizations. [34]

2.6.2 Spatial Data Analysis

It is difficult to define spatial data analysis however; many researchers have defined spatial data analysis as [53]:

"A general ability to manipulate spatial data into different forms and extract additional meaning as a result"

The presence of spatial object in data is essential for spatial data analysis [53]. Spatial data analysis allows understanding of the real world processes by developing and applying analysis criteria [52]. The underlying trend in spatial data is shown by applying these criteria [52]. Spatial analysis also allows the better understanding of hidden information within data [52]. In addition, new or unidentified relationships between the datasets can also be analyzed using spatial analysis.

Spatial analyses involve two types of operations and are given in the following [52]:

- Attribute query
- Spatial query

Attribute query selects the attribute information of spatial data. In other words, it is a process of asking logical question by selecting information [52]. For example, each road segment is represented by different identification number in a city database. A simple attribute query is to select the identification number of specific road.

Spatial query involves the process of selecting the spatial object based on location or spatial relationships [52]. Spatial queries are supported by spatial databases. For example, in spatial database it might be asked to select the number of point objects existing within polygon objects.

Using both attribute query and spatial query can result in complex spatial relationships [52]. However, these relationships can be made simple by using a method of exploratory data analysis. The exploratory data analysis displays the spatial data on maps which helps to understand the complex patterns within the data. In addition, it identifies the data properties by visualizing the data on map.

3. METHODOLOGY

This chapter explains the approach used in this thesis to fulfil the objectives. The chapter illustrates different approaches and their usability researched during the course of this thesis. In addition, the chapter justifies the use of the applied approach.

3.1 Approach

Familiarity of previously given background is necessary to understand this chapter. To fulfill the objectives, approach of thesis is divided in two parts:

- Converting the contract data (maps, requirements) and machine data in the digitalized form.
- Definition of Done is shown with the help of digitalized data.

The first step converts the contract data in the digitalized form. The contract data contains the contract area maps, inventory information (roads and equipment lists) and work requirements. The work requirements exhibits the requirements needed to complete the work whereas, the contract area maps show the roads that should be maintained. In addition, maps also illustrate different features that will affect the road maintenance work. Maps and work requirements are not given in the form, which does not support the aim to automate resourcing and tracking of work progress. Therefore, the contract data is converted into digitalized form. In Figure 11, it can be seen that first step converts the contract data comprising of maps and requirements into digitalized data.

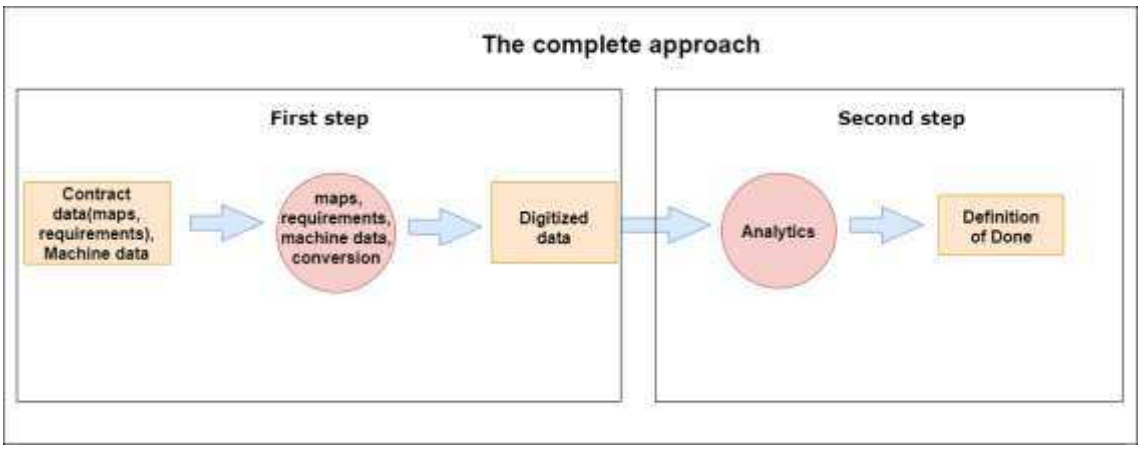


Figure 11: The complete approach

In first step, machine data is also converted into digitalized form because work requirements needed to complete the work are dependent on the machine data. In Figure 11, it can be seen that machine data is also converted into digitalized form.

In second step, Definition of Done is shown. The Definition of done is shown by using the digitalized values obtained in first step. The Definition of Done is defined as:

“A table that evaluates the completed work in a region using the digitalized machine data, maps and work requirements”.

3.1.1 Approach For Digitalizing Data

The process of converting contract data in the digitalized form is illustrated in the Figure 12. The maps from the contract data are extracted. The maps are converted into GeoJSON using an external tool. The converted file contains the representation of roads as linestring geometry along with the maintenance classes of roads. The maintenance classes of roads are classified as I, II, III. The quality of road maintenance depends upon the maintenance classes of roads.

The maintenance of roads also depends upon different road features that can affect the work. These road features include bus stops, barriers and so on. These road features are extracted from external data source. This external data source is known as DigiRoad.

The DigiRoad is a street and road database of Finland that provides all information about the traffic [66]. In addition, DigiRoad also provides centre line geometry of streets and roads [66]. It represents all the roads of Finland in the form of linestring geometry. The Finnish Transport Agency (FTA) is responsible for maintaining the database. Moreover, it also contains the attribute data of roads. The attribute data contains different values in the digitalized form. These values show the properties of road features. For example, the direction of bus stop is shown with the help of attribute data in the form of 0, 1 and 2.

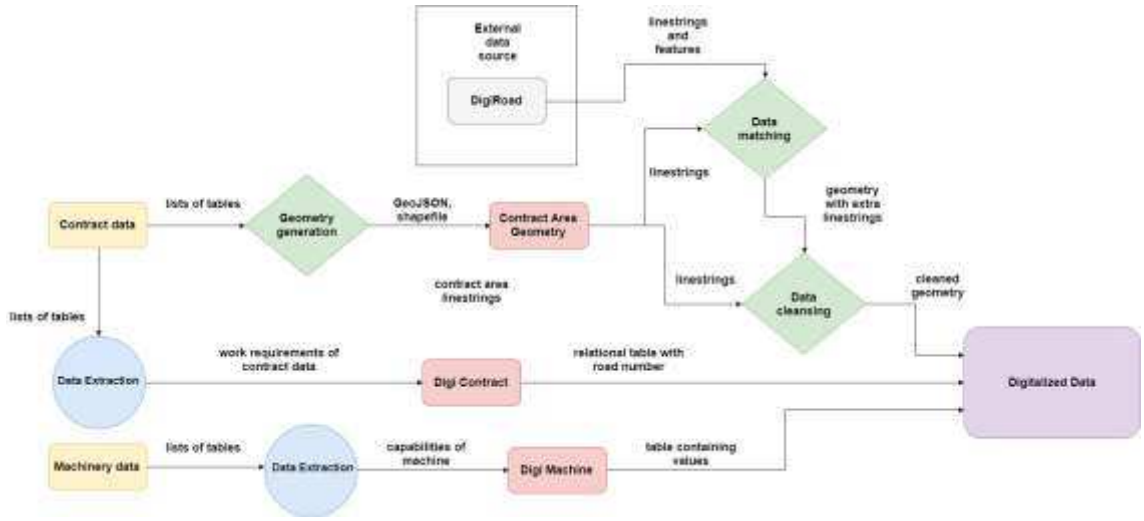


Figure 12: Digitalizing Data

The DigiRoad geometry not only represents the similar roads in map geometry but it also contains extra streets. Both geometries are also different because DigiRoad is updated on regular basis. The road features are also attached to the geometry of DigiRoad. Therefore, to get the features of roads, one approach could be to match both geometries which will results in a single geometry. One of the advantages of this approach is that it reduces the redundancy due to use of single geometry.

The matching requires selection of either DigiRoad or contract area map geometry. Selecting geometry of contract area map as a result of matching would not produce right results. This is because geometry of contract area map is not updated regularly. For desired results, the need is to use more mature data source.

Selecting geometry of DigiRoad as a result of matching would produce right results because it is maintained regularly by FTA. Therefore, any changes in real road environment will be visible in DigiRoad. To use the geometry of DigiRoad, there is a need to select DigiRoad geometry similar to contract area map geometry.

The coordinate system of DigiRoad and contract area map geometry is different. The matching requires both geometries to have same coordinate system. The matching is done to remove the differences between geometries. The matching between the geometries will select the DigiRoad geometry similar to contract area map geometry. The geometry produced by matching is called matched DigiRoad geometry.

The DigiRoad and contract area map geometries are so differently represented that matching will result in extra linestrings in matched DigiRoad geometry. These linestrings represent streets in real world. These extra streets are of no importance because maintenance work on these streets would require extra resources. Moreover, these linestrings are also not included in original contact map geometry. Therefore, these linestrings should be removed from matched DigiRoad geometry.

Linestrings are eliminated using the technique of data cleansing. The cleansing of linestrings requires the reference. The reference will define that whether extra linestrings should be removed or not. The original contract area map is taken as the reference as shown in the Figure 12. The extra linestrings are cleaned by matching each linestring of matched DigiRoad geometry with the nearby linestring of contract area map geometry. This will result in a geometry which is represented similar to the map geometry.

The contract area data also contains the work specifications and requirements. These specifications and requirements are converted to digitalized form. The information is extracted from contract data and converted into digitalized form. This process is shown in Figure 12. In this thesis, the digitalize work specification and requirements are called as DigiContract.

The DigiContract is created by adding all the requirements in the relational table but this approach is not appropriate. This is because when features of roads are changed the requirements to clean the area will also change. The requirements are closely related to the features of road. Therefore, DigiContract should contain the road information. The road identification number is extracted from DigiRoad. This extracted identification number is placed adjacent to each requirement in the DigiContract. Therefore, once the road feature is changed the requirement to complete the work will also change.

To perform the required cleaning operation, another data source is needed. This data source is called as machine data. The machine data contain all the specifications of the vehicle used in road maintenance such as weight, cleaning-capacity, type, height, width and other capabilities. This data is converted into digitalized form. The Figure 12 shows the process to convert the machine data in the digitalized format. In this thesis the digitized machine data is called as DigiMachine

3.1.2 Approach To Create Definition of Done

The Definition of Done is created by using digitalized data. The digitalized data contains the geometry similar to contract area map, DigiMachine and DigiContract. The Definition of done will evaluate the completion of work in a region. For example, how many runs it would take in a certain region to justify that work in an area is completed. In addition, the Definition of Done will also evaluate that work requirements in contract data are fulfilled. The work in an area will only be considered as done if it fulfils the contract data requirements.

The Definition of Done is shown as a table containing digitalized values. These values will evaluate the work according to work requirements. These values are obtained by creating different formulas. The formulas will generate the digitalized values depending

upon the inputs. Therefore, if the inputs to the formulas are changed the digitalized values in the table will be changed.

Data sources such as DigiContract, DigiMachine and matched DigiRoad geometry are used to create formulas for Definition of Done. The Definition of Done cannot be estimated by using a single data source because single data source does not show the complete view of maintenance work. For example, matched DigiRoad geometry only shows the geometry and features of road. Similarly, DigiMachine and DigiContract show capabilities of machines and work requirements.

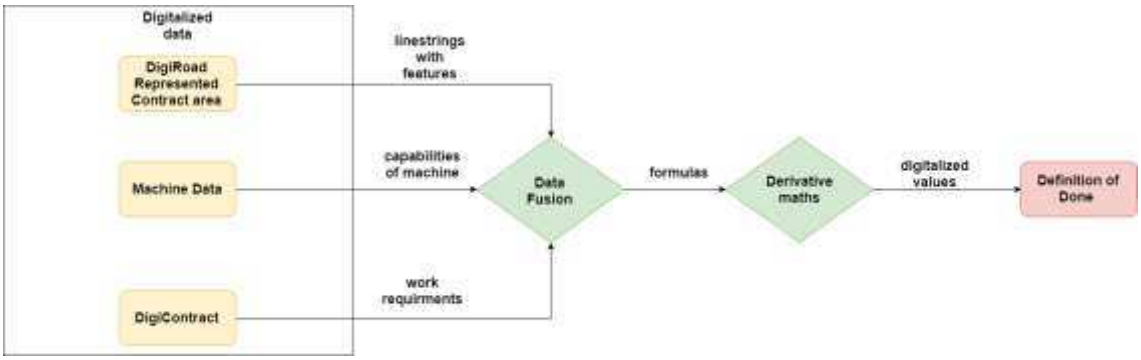


Figure 13: Process to create Definition of Done

These all data sources are combined by using a technique known as data fusion. The Figure 13 show the whole process of estimating the definition of done. As shown in Figure 13, the data sources are fused together to give a combined result. Once data sources are fused different techniques of mathematics and physics are applied to create the Definition of Done. The mathematics and physics will be used to create different formulas. The formulas will generate the values in the digitalized form. These values will evaluate that work in a region is completed according to the contract data requirements.

4. IMPLEMENTATION

This chapter describes the implementation of selected approach to achieve automated estimation and tracking of work progress. In this chapter different technologies are used to implement the solution. Moreover, this chapter explains the flow of data between different components by presenting a sequence diagram.

The selection of database in thesis depends upon various factors. The major one of them is that it should support both spatial and non spatial data. Moreover, the selected database should also fulfill the future needs. There are two options to select the database and are given in the following:

- PostGIS
- SpatiaLite

PostGIS is suitable for the type of application when multiple people are writing and reading the data from the database. In contrary, SpatiaLite is best choice for the application if a single person is reading and writing the data. SpatiaLite contain the single file which makes it more shareable as compared to the PostGIS. SpatiaLite is also easy to use in mobile applications because single file can be embedded in the mobile device. In addition, SpatiaLite also uses SQL to perform different spatial and non spatial operations which makes it easier to use. Therefore, SpatiaLite was selected as the database in this thesis as only a single person is reading and writing the data in database.

The implementation of this thesis is shown in Figure 14. The contract area map geometry is inserted in SpatiaLite database. The contract area map geometry is matched with DigiRoad geometry to get the road features affecting the road maintenance. This results in DigiRoad geometry containing extra linestrings. These linestrings are removed using the Python module. This results in DigiRoad geometry similar to contract area map geometry. This geometry is called as matched DigiRoad geometry.

The Definition of Done is also created with the help of Python module. To create Definition of Done, work requirements are digitalized. The work requirements are also dependent on the capabilities of machines. Therefore, capabilities of machine are also digitalized. The digitalized work requirements, capabilities of machines and matched DigiRoad geometry are combined to create formulas which will generate values for Definition of Done table. The following sections describe the techniques in more detail.

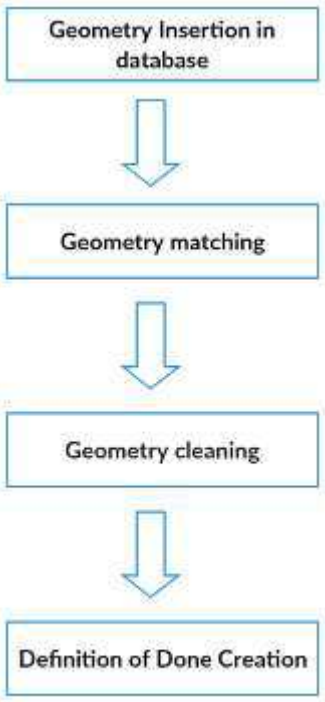


Figure 14: Implementation of thesis

4.1 Geometry Insertion in Database

To implement the proof of concept a contract area map has been chosen. The contract area map is given in different formats. The Portable Document Format (PDF) map of a contract area is shown in Figure 15. In Figure 15, the maintenance classes of roads are illustrated with different colors given at the bottom.

The contract data contain the tables to show the lists of roads in the area. These lists of roads are represented as GeoJSON data by using an external tool. The GeoJSON data contain linestrings and coordinates in the form of longitude and latitude. To add GeoJSON data in the database, an SQL query is written. The query uses a simple insert operation and a SpatiaLite built-in function known as GeomFromGeoJson. The GeoJSON data is inserted in database table. The table contains the unique primary key along with the road represented as linestring.

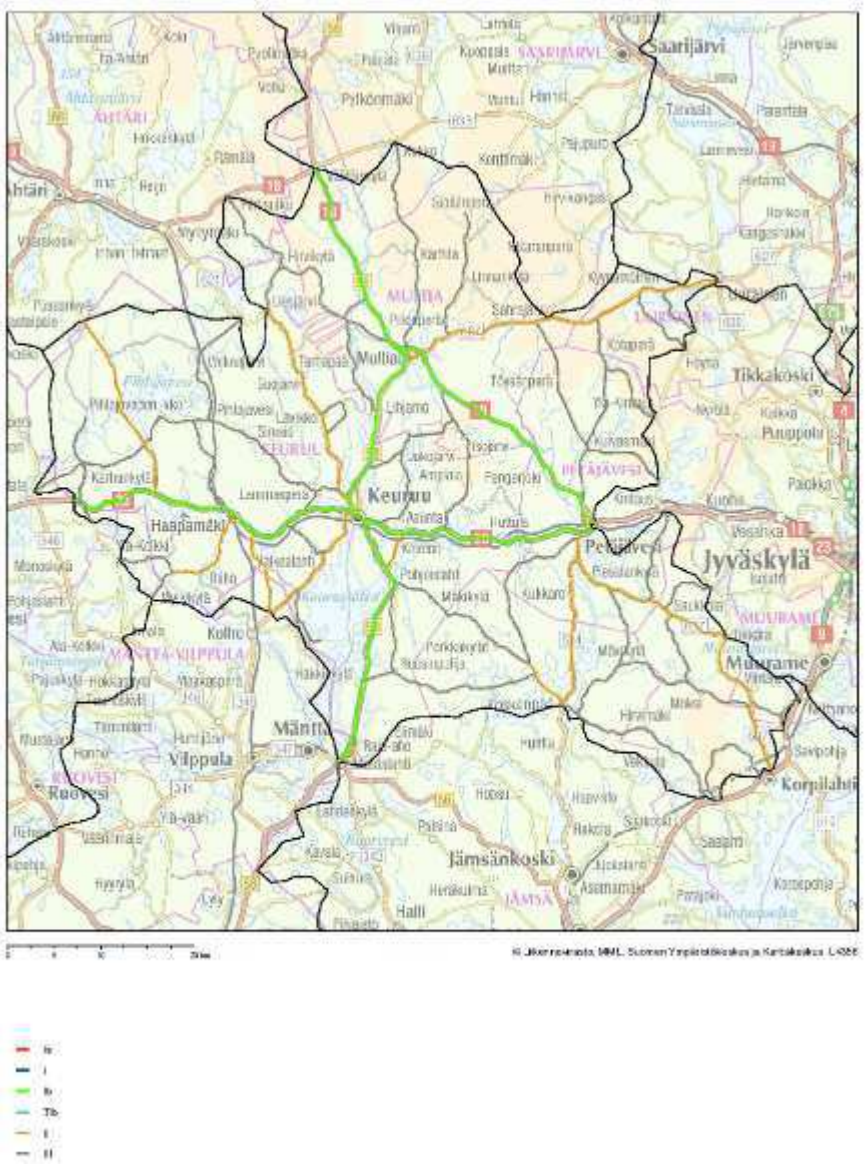


Figure 15: Contract area map in PDF

The inserted data is analyzed by an external tool to ensure that inserted data contain all linestring segments of contract area map. The data analysis tool used in thesis is known as Quantum GIS (QGIS) [72]. QGIS is open source software that allows the analysis, viewing and editing of spatial data. QGIS also provide many other functions to manipulate spatial data but in this thesis it is only used for visualization and analysis. The view of contract area map is shown in the Figure 16. Different linestrings representing roads are shown in Figure 16. This area is the representation of the Figure 15 given in human readable format.

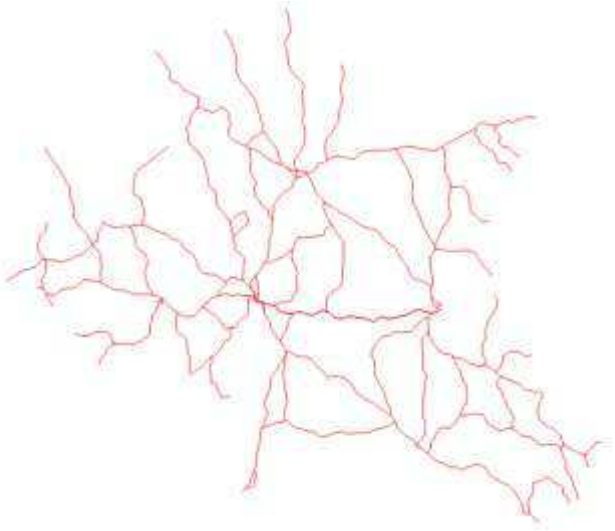


Figure 16: Contract area map as linestring

4.2 Geometry Matching

The contract requirements are bind to the road geometry and features. Therefore, there is a need to extract the features from DigiRoad. The geometry in DigiRoad is represented as linestrings. Moreover, it also contains different properties of roads. Each property is placed in its own shape file. For example, the road width is placed in the shape-file "DR_LEVEYS". Similarly, the geometry of the road is placed in a separate shape-file known as "DR_LINKKI". These files should be added in database in a similar way as the contract area map geometry is added. The portion of DigiRoad is shown below in the Figure 17. The Figure 17 shows interconnected network of roads in DigiRoad. All linestrings in the Figure 17 represent streets of different types. The red dotted line in Figure 17 shows the contract area map streets inside the DigiRoad geometry. Figure 17 show that DigiRoad geometry also contains the extra streets which are not part of contract area map geometry.

In order to utilize the features in DigiRoad, the contract area map geometry should be matched with the DigiRoad geometry. In other words, the aim is to replace contract area map geometry with the portion of the DigiRoad geometry corresponding to the contract area map. As a result, the road network features in DigiRoad can be now accessed for contract area map.

To match both geometries, geographical coordinate system of contract area map geometry is changed. The contract area map is given in the form of WGS-84 which is global coordinate system. The DigiRoad is given in ETRS-89 which is local coordinate system of Finland. The coordinate system is changed by using the built-in functions of SpatiaLite. Transform and SetSrid functions are used to convert the coordinate system

of geometry. The conversion of coordinate system will place the contract area map geometry close to DigiRoad geometry. However, both geometries model roads with certain accuracy and therefore, they slightly differ. In addition, geometries are separated by gap because the contract area map geometry is extracted from different source and perhaps at different time. Moreover, update policies between geometry sources might be different. The benefit of DigiRoad is that it is, as a national road network model, regularly maintained by FTA.

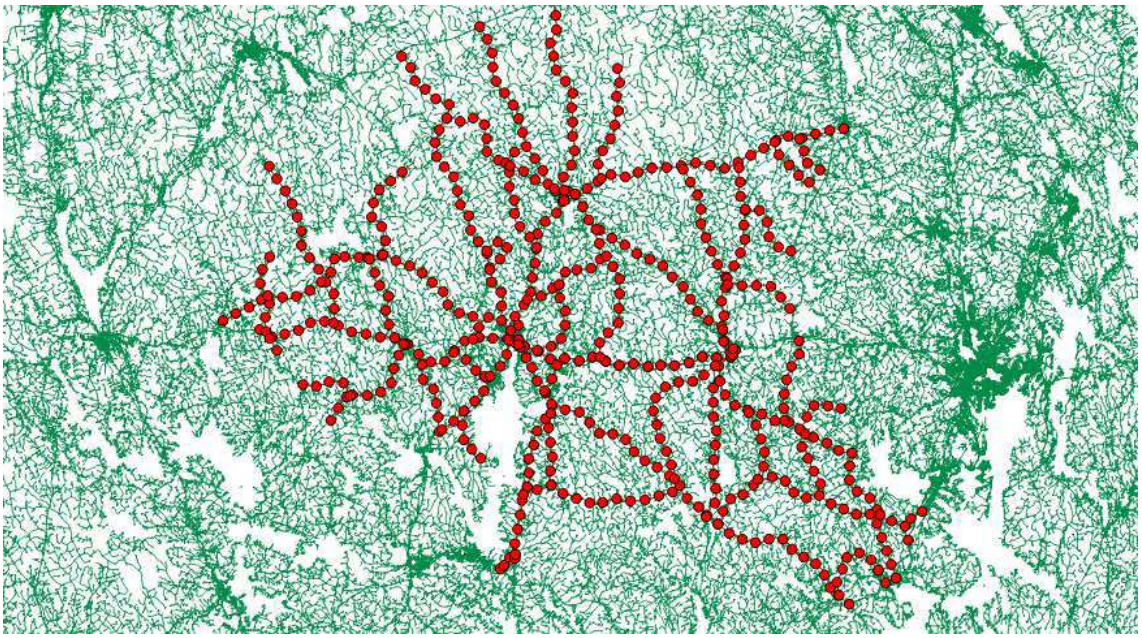


Figure 17: DigiRoad geometry

The difference between two geometries is shown in the Figure 18(a) and Figure 18(b). In both cases green geometry represents the geometry of DigiRoad whereas the red geometry is represented as contract area map. In Figure 18(a), it can be seen that geometries represents the same road however, the two geometries are different and an extra street is also present in the DigiRoad geometry. In Figure 18(b), a similar intersection is shown in both geometries. It can be seen that both geometries have a gap between them. Therefore, to extract the features both geometries are matched and DigiRoad geometry similar to contract area map geometry is selected.

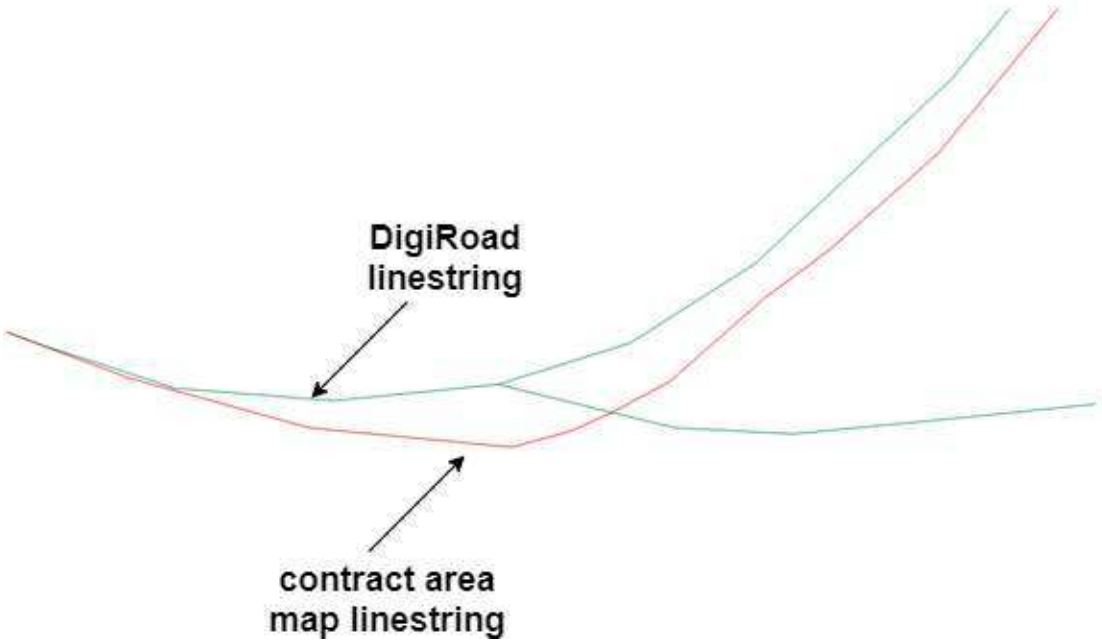


Figure 18(a): Difference between DigiRoad linestring and contract area map linestring

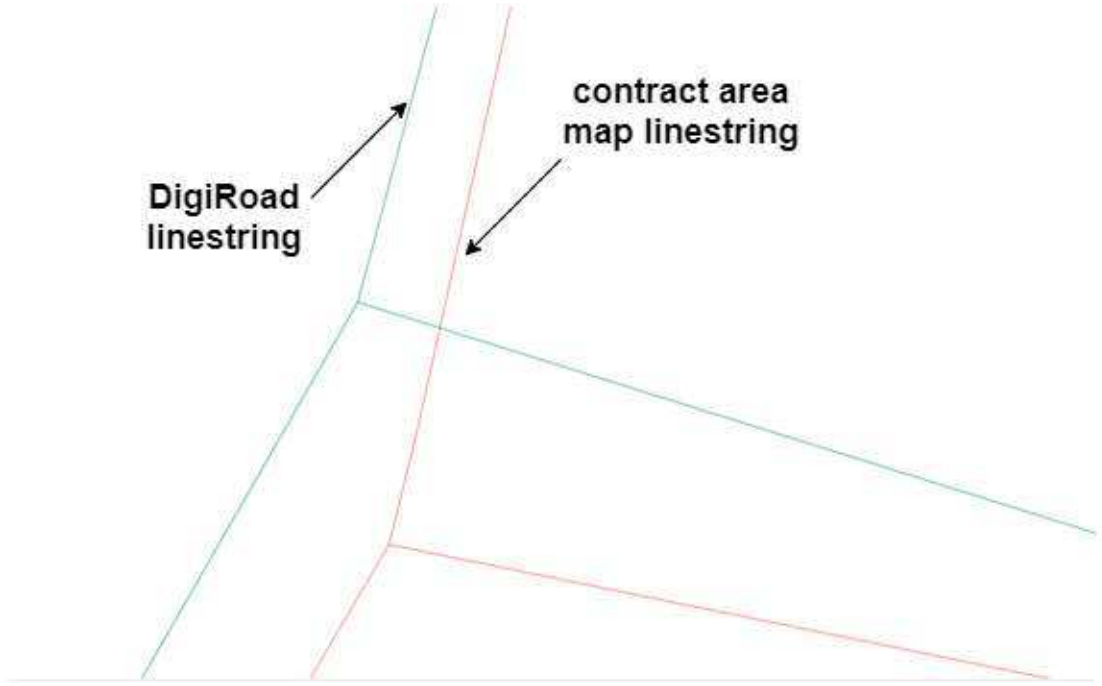


Figure 18(b): Difference between DigiRoad linestring and contract area map linestring

Buffer intersection method is used to match contract area map geometry and DigiRoad geometry. This method will select the DigiRoad geometry similar to contract area map geometry. Contract area map geometry is extended by creating a buffer. Buffer is created by using a SpatiaLite function named as ST_Buffer. ST_Buffer takes a constant value as a parameter and creates a bounded polygon similar to the shape of the geometry.

An example of ST_Buffer is shown in the Figure 19. ST_Buffer is used to extend one of the geometry in such a way that other geometry is enclosed in it. In Figure 19, the blue linestring is placed inside green buffer.

SpatiaLite function named as ST_Within is used to match both the geometries. To use ST_Within a SQL query is written in SpatiaLite. ST_Within documentations states that Boolean true will be returned if first geometry is completely inside the second geometry.

ST_Within is applied on the buffer representing contract area map geometry and on the linestrings representing DigiRoad. However, ST_Within does not match both geometries because some of the linestrings present in contract area map geometry are missing from the resulting DigiRoad geometry. This is because ST_Within wants each and every part of DigiRoad linestring segments to be inside the buffer of the contract area map geometry. The geometries of contract area map and DigiRoad are different. It is not possible that DigiRoad geometry is completely inside the contract area map buffer.

Figure 19: Example of ST_Buffer

The matching between geometries is possible by selecting complete linestring of DigiRoad if a small portion of DigiRoad linestring is inside the contract area map buffer. The query is written by using a SpatiaLite method named as ST_Intersects. ST_Intersects states that Boolean true will be returned if small portion of first geometry intersects with the second geometry.

By using ST_Intersects a complete geometry of DigiRoad similar to contract area map is selected but also the resulting geometry contain some extra linestrings. Extra linestrings are added in the geometry because ST_Intersects includes a complete linestring from the DigiRoad if a small portion of both geometries intersects.

The length of these extra linestrings is large. The length of these linestrings can be reduced by including only those linestrings which are enclosed inside contract area map buffer. Therefore, the next query involves the combination of two SpatiaLite functions named as ST_Intersection and ST_Intersects.

ST_Intersection will return the geometry which is enclosed in the contract area map buffer and ST_Intersects will find the linestrings that are in contact with the buffer. The ST_Intersection and ST_Intersects are applied on the buffer of contract area map geometry and DigiRoad linestrings.

In Figure 20, DigiRoad linestring is colored as black and contract area map buffer is colored as brown. The Figure 20 shows that small portion of black DigiRoad linestring is inside the contract map buffer. Therefore, the selected geometry will contain these small linestrings. The geometry selected by using ST_Intersects and ST_Intersection is called as matched DigiRoad geometry. These linestring are extra roads that should be removed because there is no need to get the features of these extra segments.

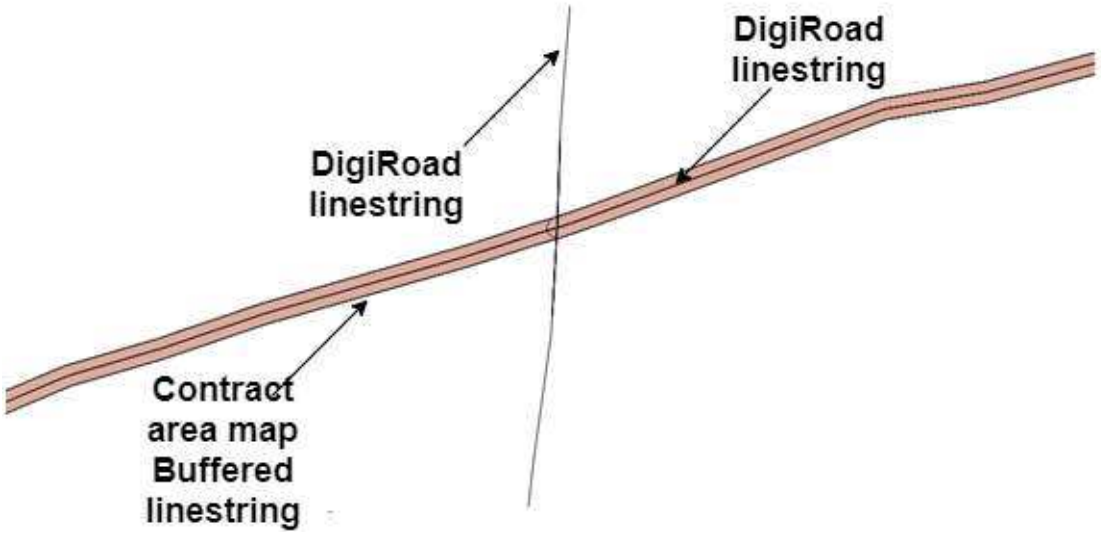


Figure 20: Intersection between contract area map buffered linestring and DigiRoad linestring

4.3 Geometry Cleaning

This section describes the approaches to clean the matched DigiRoad geometry. Geometry of matched DigiRoad geometry is cleaned by using different SpatiaLite methods. However, SpatiaLite methods does not provide the desired results therefore, extra linestrings are removed by using Python.

4.3.1 Using SpatiaLite Methods

To clean the geometry, extra linestrings are identified from matched DigiRoad geometry. The original contract area map geometry is used as reference. The reference will decide about the inclusion or elimination of linestrings from matched DigiRoad geometry. To identify extra linestrings a SpatiaLite function named as ST_Difference is used.

An SQL query is written using ST_Difference. ST_Difference returns a geometry that represents the part of first geometry that does not intersects with second geometry. ST_Difference does not produce the desired results because it requires that both comparing geometries should be strictly over each other but in this case both geometries are different and a gap exist between them.

The length of extra linestrings in matched DigiRoad geometry is very small. Therefore, length of each linestring is compared with the threshold value. The linestring is removed if length of linestring is less than threshold value. This method walks on each linestring of matched DigiRoad geometry and finds the length of each linestring by using a method named as ST_length. However, this approach has not produced the desired results because some segments in matched DigiRoad geometry have length less than threshold but these segments are present in the original contract area. Using this method, these segments will also be removed along with the extra linestring segments in matched DigiRoad geometry.

Each extra linestring in matched DigiRoad geometry creates an intersection point. These extra points in matched DigiRoad geometry can be removed by comparing intersecting points of both geometries. Each intersecting point in contract area map geometry is compared with the intersecting point in matched DigiRoad geometry. As both geometries represent similar roads therefore, both geometries will have same intersecting points. However, if the intersecting point in matched DigiRoad geometry is not present in contract area map geometry this means it is an extra point and should be removed.

Removing the extra intersecting point will remove the extra linestring. However, an intersecting point might have two linestrings. One linestring is extra and other linestring is present in original contract area map geometry. Removing this kind of intersecting point would remove the linestring which is present in original contract area map geometry. Therefore, this method has not produced the desired results.

After using different built-in methods of SpatiaLite database it is concluded that SpatiaLite does not provides any built-in function to clean this geometry. To solve this problem each linestring in matched DigiRoad geometry should be compared with the corresponding linestring in contract area map geometry. This cannot be achieved by using built-in functions therefore, there is a need to write custom function in SQLite. However, SQLite does not provide the functionality to write custom functions. Similarly, SpatiaLite is built on top of SQLite therefore, it also does not allow writing the custom functions. Therefore, there is a need to use a programming language to clean the extra linestrings from matched DigiRoad geometry.

4.3.2 Using Python

Python is used in this research work to clean the extra linestring from matched DigiRoad geometry. Python have made the programming easy and quick. Moreover, Python has also provided many standard libraries to manipulate spatial data which are not present in other programming language. The most commonly used libraries are shapely and Geospatial Data Abstraction Library (GDAL).

To clean the geometry each linestring in contract area map geometry is compared with matched DigiRoad geometry. The length of linestrings in both geometries is not same. The length of contract area map linestrings is large as compared to the length of DigiRoad linestrings. Using shapely and GDAL, a Python function is written that divides the DigiRoad linestring in smaller portions. The Python function takes the breaking length of linestring as a parameter and breaks the contract are map linestring in portions. In addition, the DigiRoad geometry is also divided into several portions using the same Python function so that lengths of the both geometries are same.

To compare the geometries, the DigiRoad linestring closest to contract area map linestrings are found. This is achieved by calculating the distance between each matched DigiRoad linestring with all contract area map linestring. The distance is calculated by using shapely library in Python. The distance between the geometries is placed in the Python list.

After calculating the distance, bearing angle is found between the matched DigiRoad linestring and all linestrings of contract area map. The bearing direction also known as azimuth of linestring gives the direction of linestring in degrees. The bearing angle is selected for comparison because the direction of all extra linestrings in matched DigiRoad geometry is different from the direction of linestrings in contract area map geometry. Bearing formula of mathematics is used to calculate the bearing angle between the linestrings of two geometries. The difference of bearing angle between the contract area map linestrings and matched DigiRoad linestring is calculated. The difference of bearing angle is also placed in Python list.

The results of bearing list and distance list are combined because linestring comparison will be based on these two properties. The bearing and distance lists are converted in the list of tuples. In Python, a tuple is a special kind of data structure that groups the number of items in a single compound value. Therefore, the tuple list converts the two separate lists into a single list. The single list now contains the distance and bearing difference between matched DigiRoad linestring and contract area map linestrings.

A threshold value for bearing and distance is set which will determine the presence of matched DigiRoad linestring in original contract area map geometry. The distance and bearing of linestring stored in a tuple list is compared with the threshold value. The

matched DigiRoad linestring is eliminated if the distance and bearing of matched DigiRoad linestring is greater than the threshold value. This method is repeated for all linestring of matched DigiRoad geometry.

This technique works and removes all the extra linestrings from the matched DigiRoad geometry. However, this technique works only when both matched DigiRoad and contract area map geometry is divided into very small portions. In addition, it is not an efficient method because it is taking more time to clean the geometry and roundabouts are also not properly cleaned. Therefore, this method might be producing the right results for simple linestrings but it is not an efficient method.

To optimize the code, Python script is analyzed and it is concluded that comparison between geometries to find the nearest contract area map linestrings is taking extra time. Therefore, this comparison cannot be done by iterating all the contract area map linestrings. To optimize the code, Python script is connected to SpatiaLite server database. SpatiaLite query uses a built-in R Tree indexing feature that compares geometries within seconds. However, to use R Tree indexing in SpatiaLite there is a need to write a sub query in the query.

A new Python script is written which connects SpatiaLite database server to code. This script uses R Tree indexing in SpatiaLite to compare both geometries. The R tree indexing finds the contract area map linestrings nearest to matched DigiRoad linestring within seconds. The contract area map linestrings nearest to matched DigiRoad linestring is found by writing an SQL query in Python code. The query uses a built-in function of SpatiaLite named as PtDistWithin. The function PtDistWithin takes constant numerical distance value within the parameters and finds the nearest linestring within the given distance value.

The matched DigiRoad linestring and contract area map linestrings are then compared with each other. The comparison is again based on the bearing angle of both the comparing linestrings. The bearing difference is calculated between the bearing angle of matched DigiRoad linestring and nearby contract area map linestrings. This difference is then compared with the threshold value. The extra linestring in matched DigiRoad geometry is eliminated if the bearing difference between the linestrings is greater than the threshold.

The roundabouts are detected by creating a Python function. This Python function checks the coordinates of each linestring. The roundabouts are detected by calculating the difference between the first coordinate pair and the last coordinate pair. If the difference between them is zero this shows that geometry is a roundabout.

A new partially cleaned geometry is obtained by comparing each matched DigiRoad linestring with contract area map linestring. The algorithm takes around 7 minutes to clean the extra linestrings from geometry. However, still there are few extra linestrings

that needs to be removed. The linestrings are not removed because the difference between the bearing angle of linestring in matched DigiRoad and contract area map is less than the threshold. One example is shown in the Figure 21. The purple linestring shown in the Figure 21 is the extra linestring that should be removed. This linestring is compared with the green linestring in the contract area map geometry. It can be seen that bearing angle (direction) of both linestrings seems to be same therefore, the difference between bearing angle is less than the assumed threshold value.

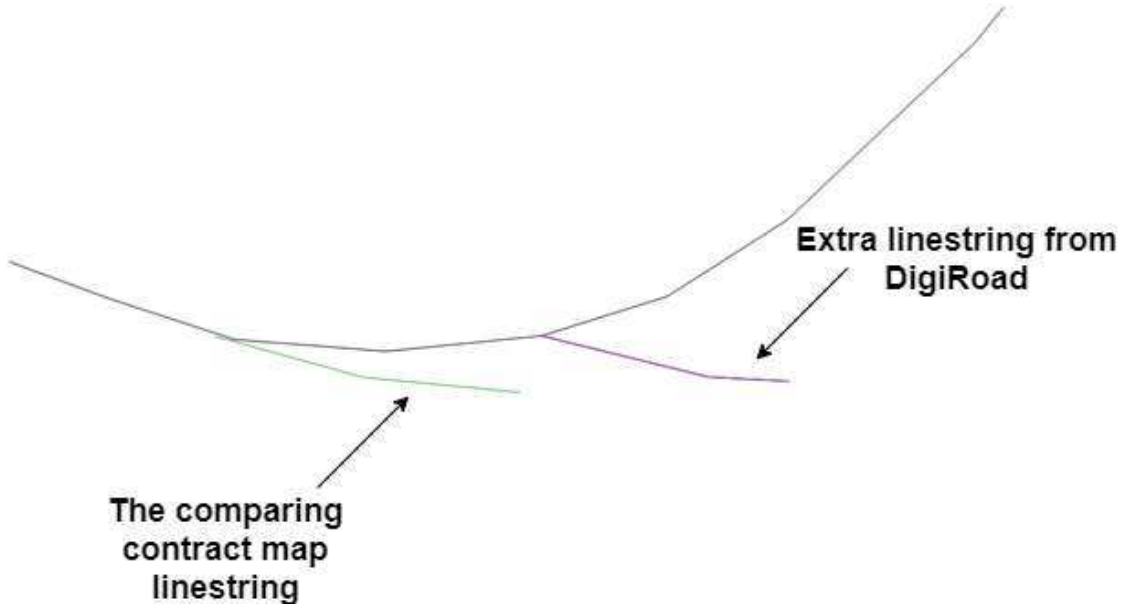


Figure 21: Linestring of DigiRoad and linestring of contract area

The geometry can be cleaned by detecting the point where extra linestrings are not cleaned. All the extra linestrings has one point connected to the linestring whereas other point is not connected to any linestring in geometry. This behavior is visible in the Figure 21. It can be seen that purple line has starting point connected to the black linestring whereas other point is not connected to any linestring. The extra linestrings are detected where more than two linestrings are connecting at a point. To detect extra linestrings within geometry intersecting points within whole geometry is found.

The intersecting point within the geometry is found by writing a SpatiaLite query using ST_Intersection. The algorithm walks on each intersecting point of the geometry and checks the linestrings connecting at that intersecting point. If more than two linestrings are intersecting at a point then starting point and ending point of these linestrings are checked. If the starting or ending point of these linestrings are not intersecting with any other linestring then it should be removed. This cleans the complete geometry and all the extra linestrings are removed. The geometry obtained after cleaning in called as cleaned DigiRoad geometry.

The cleaned DigiRoad geometry is then included in the SpatiaLite in the form of table. The features of roads are extracted from different tables of DigiRoad. The features are

extracted from different feature tables using SQL join. The cleaned DigiRoad geometry is joined with the feature table using identification number of road.

4.4 Creating Definition of Done

The next step is to create the table for Definition of Dones. However, to make Definition of Done the work requirements in contract data should be digitalized. Therefore, the information from the contract data is extracted and converted in the digitalized form. The digitalized work requirements are placed in a table known as DigiContract. Some of the requirements included in contract data are given below

- The requirements for machinery to work in an area
  - E.g., snow removal machines should also contain deicing capability
- The roads that require special attention in an area
  - E.g., the road such as highway 18 or regional road 604 and so on.
- The road sections where only sand is allowed for deicing
  - E.g., the main road number 58 25 6123- 7664 requires sand for deicing
- Machinery restriction on maintenance operations and classes
  - E.g., tractors are not allowed for snow removal or deicing on functional class I regional roads, functional class II main road and functional class III local main street
- Dangerous downhill's
  - E.g., regional road 348 road section 8, regional road 607 road section 3 and regional road 6071 road section 2

The DigiContract table should also view the changes in road environment because the requirements in contract data are bound to the road environment. For example, dangerous downhill is bound to road number 3048. The changes in road environment can be visible in DigiContract by storing the identification number of road in the DigiContract table. The identification number of road is found by searching the road number give in work requirements from DigiRoad. This gives the identification number of road and it is then placed adjacent to each requirement in DigiContract table.

The work requirements are also dependent on the type of operations. For example, deicing operation is required to complete the work. A separate operational table is made in database that relates to the work requirements table using the operational code column. Therefore, once the operation in operational table is queried with the help of SQL, DigiContract will show all the requirements stored for that operation.

The machine data is also formulated needed for creating Definition of Done. The machine data also known as DigiMachine is a database table that contains the working machine width, height, cleaning capacity, maximum speed and deicing capability. The machine data is formulated after extracting the information of maintenance machines.

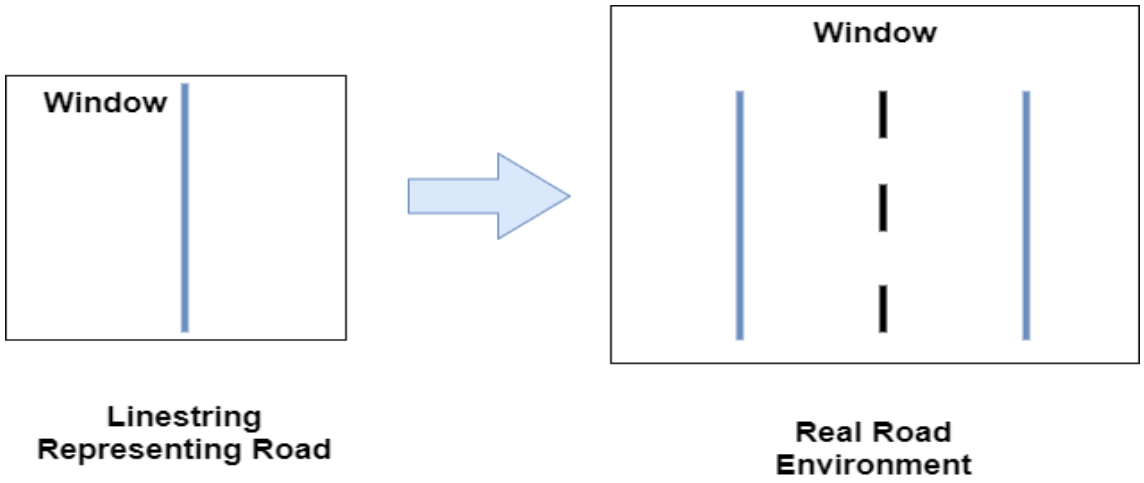
To create the Definition of Done, it is checked that work is completed in a road segment according to work requirements. Each linestring in the contract area map corresponds to the road in the real world environment. Each road segment is checked to justify that work has been done. Once work in each segment is completed therefore, it would mean that work in whole area is completed. In this thesis, Python is used again to verify that work is done in a segment according to the work requirements.

The types of roads will play an important part in showing Definition of Done. There are different types of roads present in the DigiRoad. The roads are divided into different types such pedestrian road, part of single carriage way and part of roundabout and so on. The direction of traffic flow is also shown as a number in DigiRoad table. The direction of traffic flow in DigiRoad is given as:

- Traffic is allowed in both directions
- Traffic is allowed against the direction of digitization
- Traffic is allowed in the direction of digitization

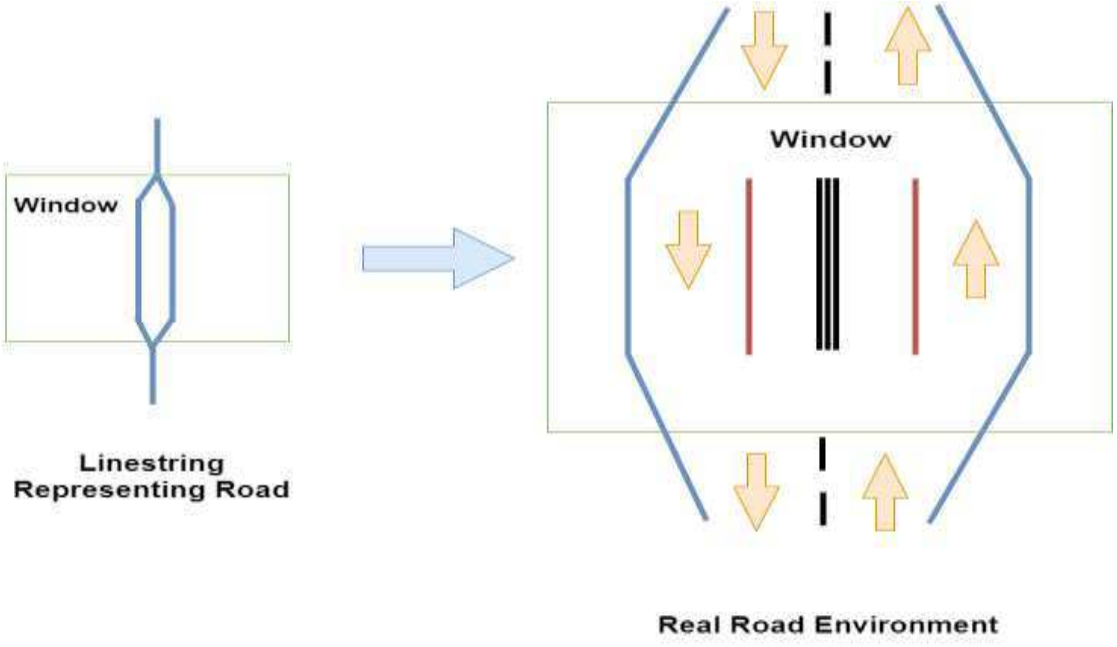
The traffic direction in digitization and against digitization is shown by a digitized value in DigiRoad. Similarly when traffic is allowed in both directions it is also shown by a digitized value. For example, to show that traffic is allowed in both directions DigiRoad has used a value of “2”.

The direction of digitization in the road segments is formulated according to the cardinal directions. The directions are formed in such a way that northern-southern road represents direction of digitization by moving from southern end to the northern end of the road whereas the direction against digitization is represented by moving from northern end to southern end in a segment. However, in case of eastern-western road, the direction of digitization is shown by moving from western end to eastern end whereas, the direction against digitization is shown by moving from eastern to western end.



(a) **(b)**

Figure 22: Window on single carriageway



(a) **(b)**

Figure 23: Window on dual carriageway

To evaluate the work done in a segment, a window is created on complete segment of road. The window will determine that work requirements to complete the work on a segment. However, to create a window there are different cases that needs to be handled in this thesis. These cases are shown in Figure 22 and Figure 23.

The first case shows a window on a linestring and its representation in real road environment. In Figure 22(a), a linestring is shown whereas, in Figure 22(b) representation of linestring is shown in real world environment as a simple two lane road with traffic allowed in both direction. The road in Figure 22(b) is also called as single carriageway with two lanes. Similarly, in Figure 23(a) a linestring is shown which is representing a road. In Figure 23(b), a real world road environment is shown in which single carriageway road is divided into two lanes separated by central reservation for each traffic direction. The road in Figure 23(b) is also called dual carriageway.

The work requirements are determined by creating a window on complete segment of single and dual carriageway. This concept is similar to dragging a window on the road and checking one by one that work in a portion is completed. The concept is illustrated in the Figure 22(b) and Figure 23(b) where a window is created on the single carriageway and on dual carriageway to verify the work done. However, to create a window there is a need to identify the single carriageway and dual carriageway. The contract area map geometry provides the information of single carriageway whereas, the information of dual carriageway is missing. An algorithm has been proposed in this thesis to differentiate dual carriageway from single carriageway.

The algorithm walks on each linestring of cleaned DigiRoad geometry. The intersecting linestrings to the current linestring is found by using SpatiaLite function named as ST_Intersects. As each linestring in cleaned DigiRoad geometry represent road, the direction of traffic flow for the current and intersecting linestrings are determined from DigiRoad table. If the direction of traffic flow for intersecting linestrings is changed as compared to the current linestring this shows that there is a possibility of dual carriageway. Therefore, if it is a dual carriageway a similar shape linestring (road) will exist in parallel to the current linestring. A window will created on the parallel linestring and the current linestring and work done is verified using the requirements. Similarly, window will be created on each linestring of single carriageway to check the work done.

The identification of the dual and single carriage way follows the process of getting the features. The features will be extracted from the feature table and will be stored in the form of dictionary in Python. In Python, dictionary is a special data structure where each key is separated by value using a colon.

The next step retrieve all the requirements attached to that linestring (road) from DigiContract. However, the requirements are linked to the operation table. Different operations have different requirements. With the help of SQL join, the requirements table will be queried through operation table. The requirements will be checked in a separate function using Python. However, to check the requirements and to create the Definition of done, DigiRoad, DigiMachine and DigiContract are fused together.

A special Python function is created to check the requirements. The window is dragged on each segment of road and requirements are checked for that segment. The road data is obtained for each segment using the DigiRoad and it is compared with the requirements in DigiContract. For example, the DigiContract contains the requirement that tractor is not allowed on class number I, II and III. Therefore, the class of each road segment is checked from DigiRoad whereas type of working machine is obtained from the DigiMachine. If the working machine is a tractor then a flag is set to show that tractors are not allowed on this segment. Similarly, DigiContract contains requirement about the roads that need special attention therefore, these roads are obtained from the DigiRoad and flag is set on each road segment that this road requires special attention. DigiContract contains another requirement that all the machines should contain the deicing capability. Therefore, the capability of the machine will be obtained from the DigiMachine and it will be checked that every working machine contains the Deicing unit. If the working machine does not contain the Deicing unit, a flag is set that it is not allowed to work. All the flags are set in the digitalized form that computer can understand. To show the requirements for each segment all the values are then added in the Definition of Done table.

Once all the requirements are checked on the complete segment, the next step is to show how work should be done in an area. For that purpose, the numbers of runs are calculated in an area to complete the work. The number of runs is calculated by creating a formula. The formula obtains the width of the segment and divides the width of segment with capacity of machine to clean the area. The width of the road is obtained from the table in DigiRoad whereas, the capacity of machine to clean the area is obtained from the DigiMachine.

The number of runs for single carriageway is calculated in a different way whereas, the number of runs for dual carriageway is calculated in a different way. In dual carriage way the number of runs is calculated by sliding the window on both parallel segments whereas, in single carriage way the number of runs is calculated by sliding the window on the single segment.

The number of runs will be affected by the bus stops present in each segment. This behavior is depicted in the Figure 24(a) with linestring representation of road with bus stop and Figure 24(b) with representation of linestring containing bus stop in real road environment. It can be seen in Figure 24(b) that the bus stop on each side of the road would mean an extra run would be required to complete that segment. However, it is also needed to find the direction of run.

To find the direction of run, a query is written in SpatiaLite to extract the direction of traffic flow from features table of DigiRoad. Once the direction of traffic is extracted there is a need to know the direction in which bus stop exists. Therefore, this information is also extracted from DigiRoad. After extracting this information, an extra run

would be added in the direction of bus stop. In this way the number of runs is calculated along with the direction of run. The number of runs will be then added in the Definition of Done table.

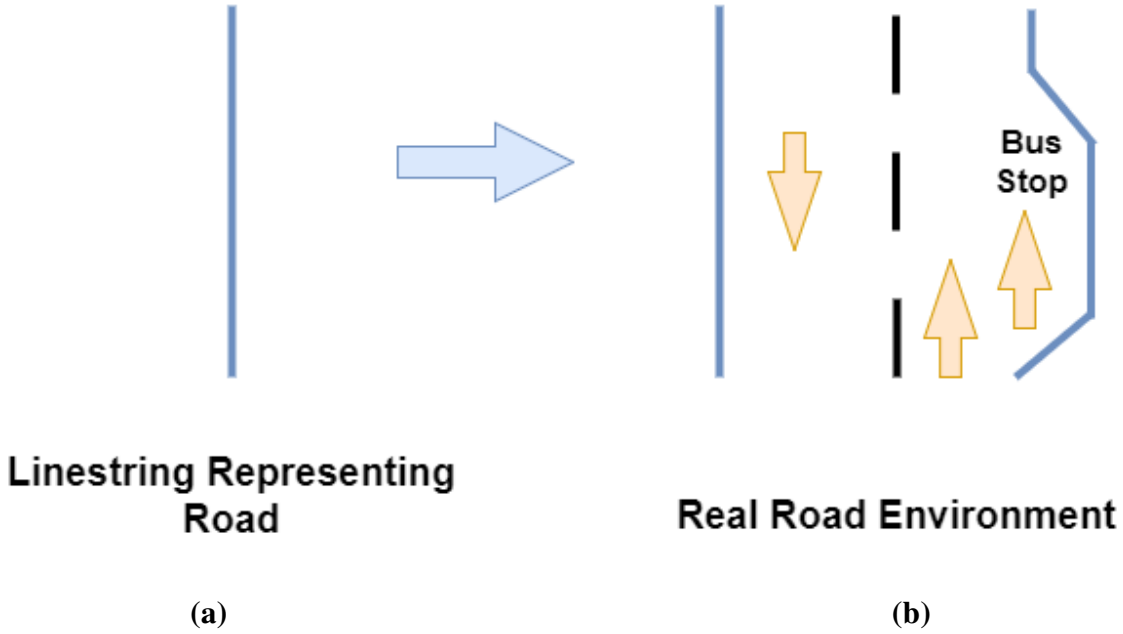


Figure 24: Bus stop on road

The table for Definition of Done contains all the data in the digitalized form. The Definition of Done table will evaluate that work is done in each segment of an area according to the work requirements.

4.5 Module Interaction

The module interaction to create Definition of Done is shown in Figure 25. The input is given in different formats and input such as shape-file or GeoJSON. The input data is inserted into database. Spatial query is applied which matches the contract area map geometry with DigiRoad geometry. The matching returns a geometry which is analyzed by a tool known as QGIS. The data analysis shows that geometry contains extra streets therefore, it requires cleansing. The cleansing of geometry is done using Python programming language. The cleaned geometry is again analyzed by QGIS. The analysis shows that geometry is partially cleaned therefore, it is again subjected to cleaning. The next phase of cleaning is again done by Python. The second phase of cleaning returns the cleaned geometry which is stored in spatial database. The cleaned geometry is created similar to contract area map. The features of cleaned geometry are then extracted using DigiRoad. All the data sources DigiContract, DigiMachine and cleaned geometry is then fused together using Python. The fused data is then further used to create Definition of Done table. The Definition of Done contains the digitalized values that show

how work should be done in each segment of an area according to the work requirements. The table is then stored in spatial database.

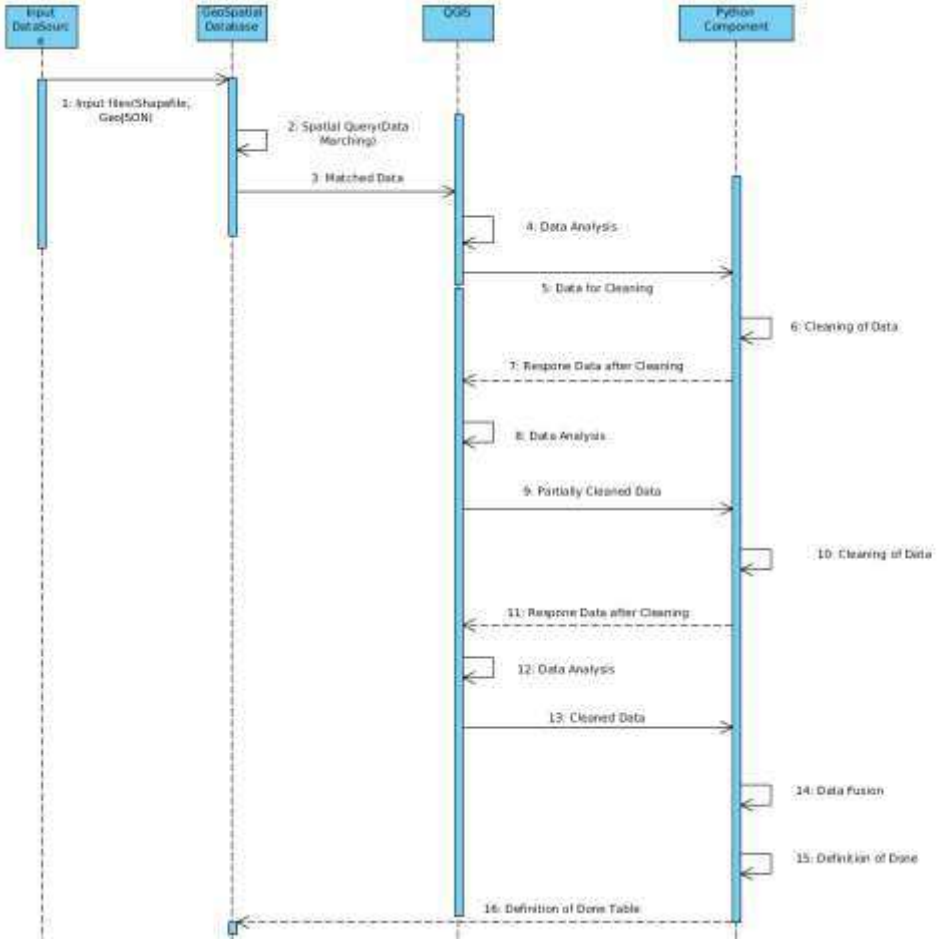


Figure 25: Sequence diagram showing complete implementation

5. RESULTS

This chapter validates the implemented technique by testing it on a use case of contract data. The chapter presents the obtained results in form of different visual diagrams. Moreover, it concludes the results in the form of table that exhibits the Definition of Done.

5.1 Visualizations and analysis

The first task is to place the contract area map geometry close to DigiRoad geometry. Therefore, the coordinate system of contract area map geometry is changed using built-in SpatiaLite functions to place the contract area map geometry close to the DigiRoad geometry. However, there exists a gap between the two geometries. The gap can only be seen if the geometries are zoomed in. This is illustrated in Figure 26.

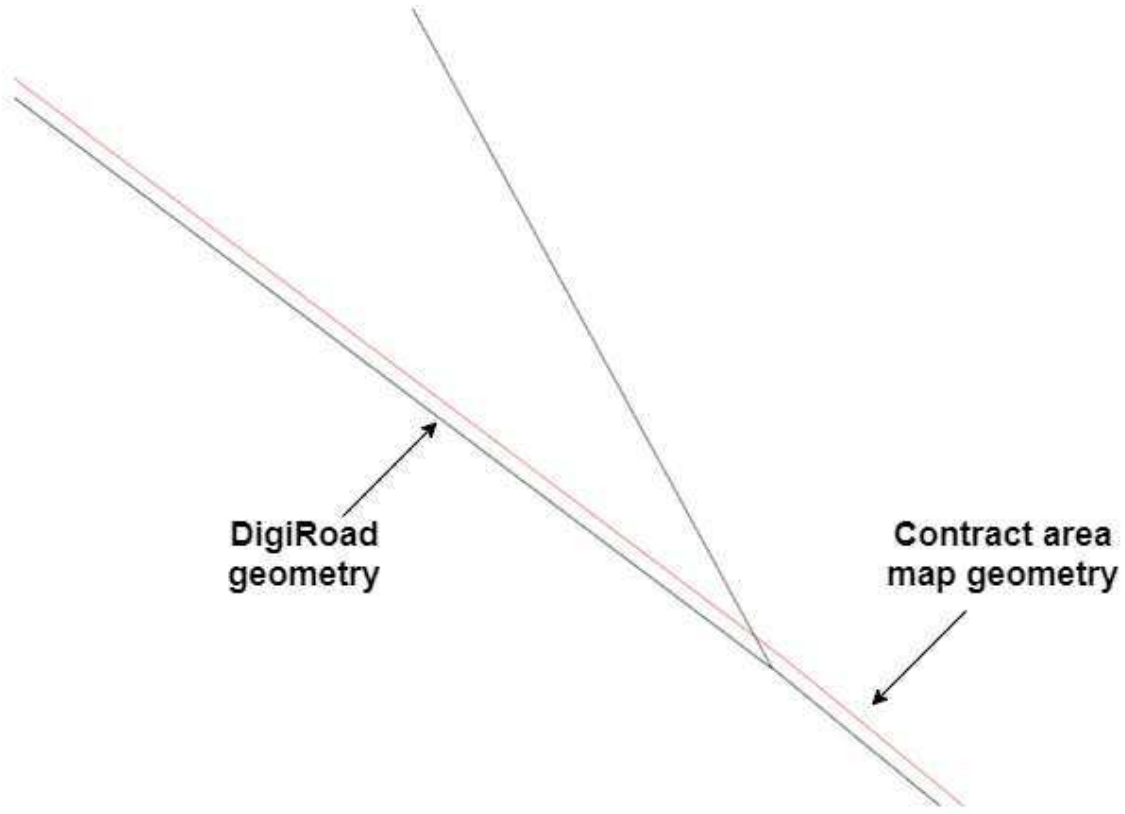


Figure 26: Result of placing contract area map geometry close to DigiRoad geometry

In Figure 26, contract area map geometry is colored as red whereas, DigiRoad geometry is colored as black. It can be seen that there exists a gap in between the geometries. To reduce this gap the technique of matching is used.

This step matches contract area map geometry and DigiRoad. The matching resulted in a geometry that contains extra linestrings. These linestrings in the geometry are small therefore, they cannot be seen if complete geometry is shown. However, the zoomed in portions of original contract area map geometry and matched DigiRoad geometry are shown in Figure 27 and Figure 28.

In the Figure 27, same roundabouts are shown from both geometries. In Figure 27, black roundabout geometry represents the matched DigiRoad geometry whereas the red roundabout geometry on the bottom represents the contract area map geometry. In Figure 27, extra linestrings can be seen which are created as a result of matching between the DigiRoad geometry and contract area map geometry. Similarly in Figure 28, identical linestring of contract area map geometry and matched DigiRoad geometry are shown. The Figure 28 represents the curvy roads instead of the roundabouts. In Figure 28, the black linestring represent DigiRoad geometry and red linestring belongs to contract area map geometry.

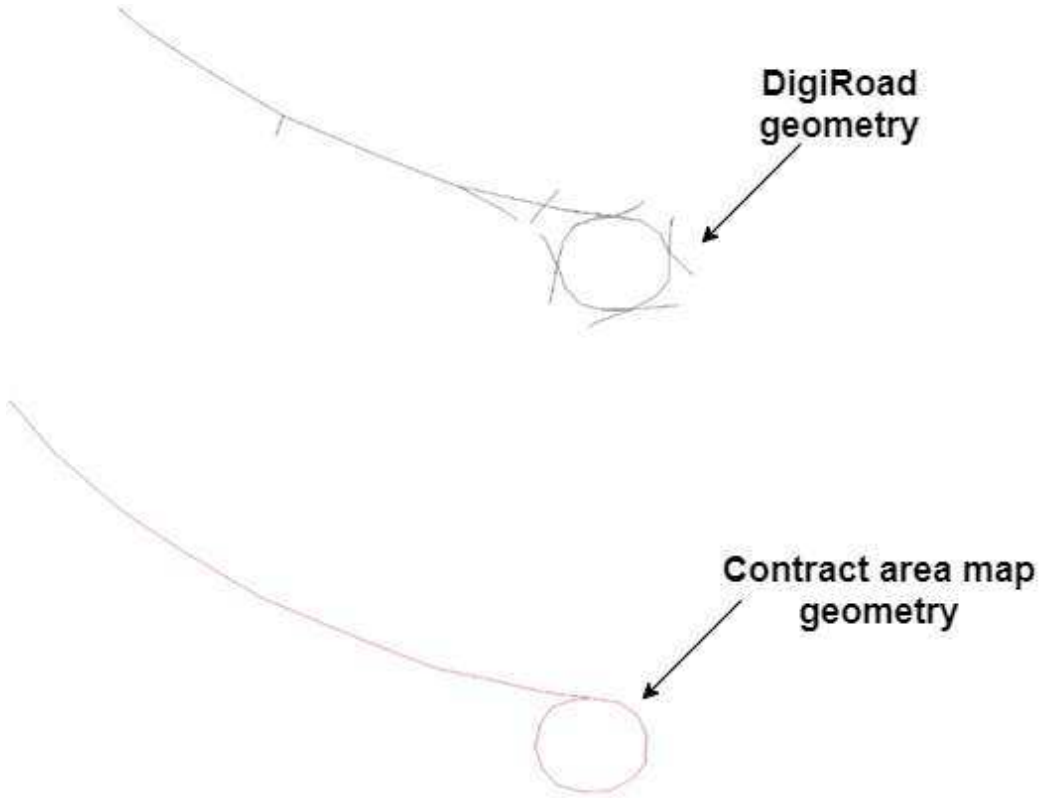


Figure 27: Roundabout result after matching contract area map geometry and DigiRoad geometry

The extra linestrings produced as result of matching is of no importance because these linestrings are not included in the contract area map geometry. These linestrings should be removed because maintenance on these linestrings would require extra resources.

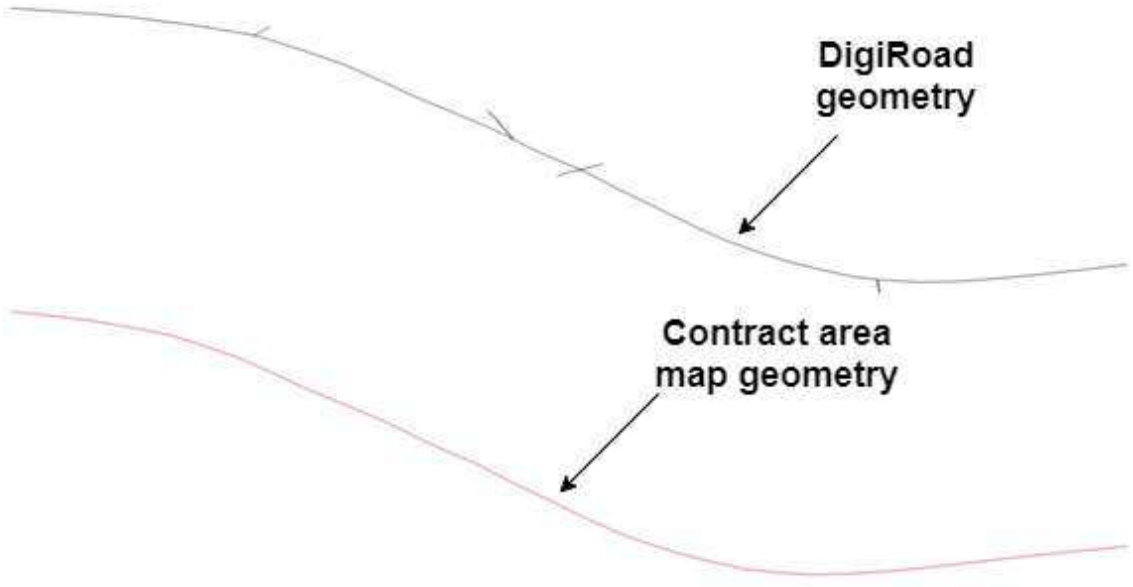


Figure 28: Curvy road result after matching contract area geometry and DigiRoad geometry

This step involves the cleaning of the geometry. The geometry is cleaned by comparing each linestring of matched DigiRoad linestring with contract area map linestrings. This results in partial cleaning of the geometry. After partial cleaning the result is shown in Figure 29.

The Figure 29 represents the cleaned area after first phase of cleaning. These linestrings are similar to the above shown curvy roads. It can be seen that most of the roads are cleaned in the area except a single extra linestring. In Figure 29, a black box is made on the extra linestring. This extra linestring should be removed to get the matched DigiRoad geometry similar to original contract area map geometry.

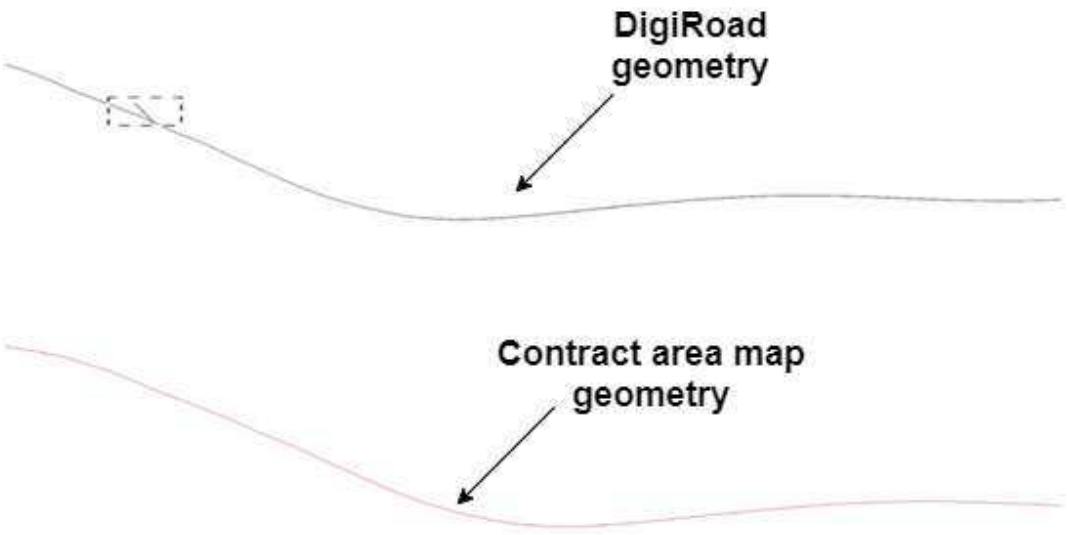


Figure 29: Result after first cleansing phase

In the next step, the second phase of cleaning is applied. The second phase cleans the matched DigiRoad geometry completely and all the extra linestrings from the matched DigiRoad geometry are removed. The area obtained from second phase of cleaning has shape similar to the contract area map geometry. However, both areas are not placed at same position. The same linestring is shown in Figure 30 as it is shown in Figure 29.

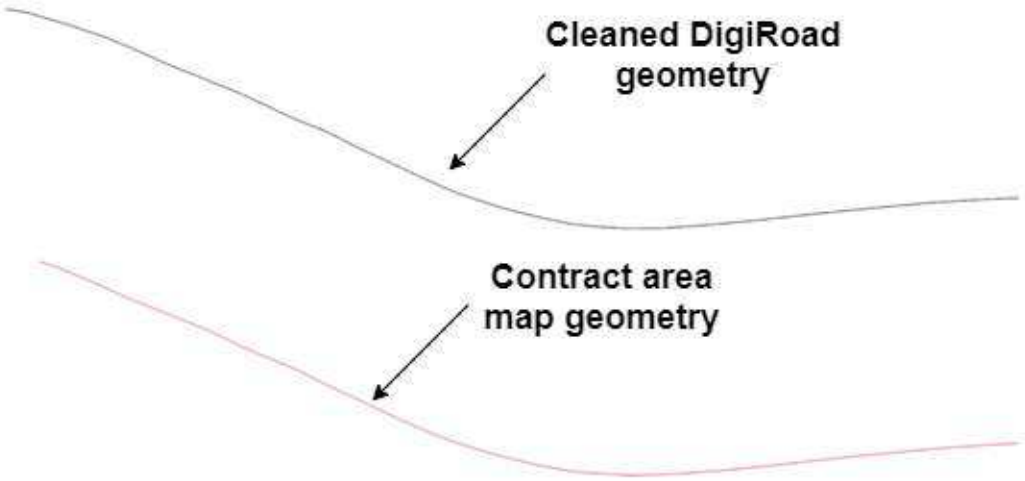


Figure 30: Result after second phase of cleansing

The next step formulates the DigiContract and DigiMachine. The DigiContract and DigiMachine contain the details that are extracted from the contract data. These details are created as SpatiaLite tables. The tables formulated are shown in the Figure 31 and Figure 32.

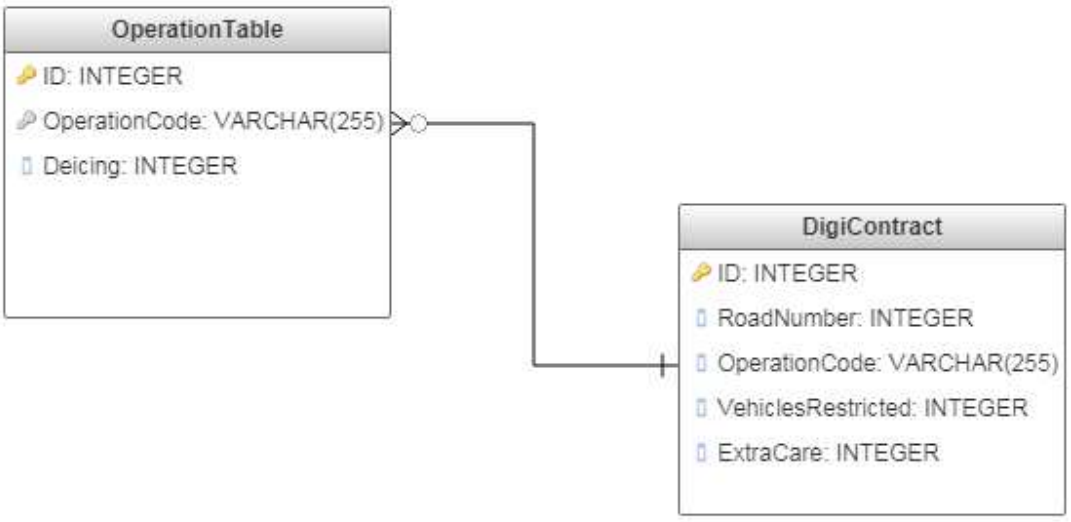


Figure 31: Operation table relation with DigiContract

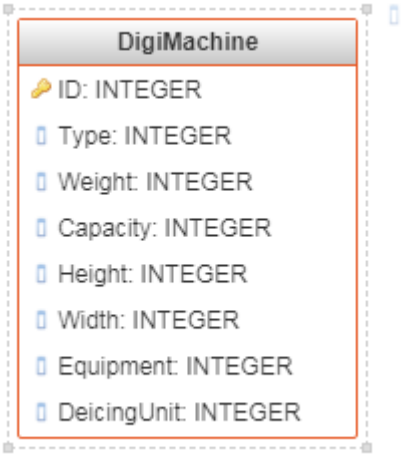


Figure 32: DigiMachine table

The Figure 31 shows the relation between DigiContract and Operation table. The Operation table is linked to the DigiContract table by column Operation Code. This is done because different requirements are needed for different operations. The Operation table also contains the Deicing column because deicing capability is required in different operations. The DigiContract requirements are accessed using the Operation Code. The DigiContract contains the Road Number which links requirement to each road segment. It also contains the information of the vehicle restricted on the road segment. Moreover, it also shows road that needs extra care. DigiContract contains all the values in the digitalized format. These values will specify different requirements on a road segment.

In the Figure 32, the table of DigiMachine is shown. It contains different machine features and capabilities. The features are extracted from real maintenance working ma-

chine. The type of the machine is specified with the help of digitalized value. The type of machines can be classified into lorry, tractor, truck, and so on. The second column in DigiMachine shows the weight of the machine. The third column shows the capacity of machine to clean the road in a single run. It is usually given in meters. The fourth and fifth column shows the height and width of machine. The next column shows what type of equipment machine contain to clean the road and last column checks that machine contains the Deicing unit.

The DigiContract and DigiMachine is digitalized which is required to automate the estimation of resources and tracking of work progress. The DigiContract shows requirements in the contract documentation whereas, DigiMachine shows the capabilities of machine which will perform the required operation. DigiContract and DigiMachine are created in the form of table which will be further used to evaluate the work done in an area.

The Definition of Done table is created by creating the formulas. These formulas are created by fusing DigiContract, DigiMachine and DigiRoad. These formulas will result in the values which will evaluate the work done in a region. The values evaluating the work are in digitalized form. The Definition of Done table is shown in Figure 33.

ID	Road_Number	Operation_Code	Extra_Care	Vehicle_Restricted	No_of_Runs
1	1213062	Snow Deicing	0	1	(4,(4,2),(3,2))
2	1213406	Snow Deicing	1	0	(5,(4,2),(3,3))
3	1213907	Snow Deicing	2	1	(9,(4,5),(3,4))
4	4098715	Snow Deicing	0	1	(4,(4,2),(3,2))
…	…	…	…	…	…
…	…	…	…	…	…

Figure 33: Definition of done Table

The Definition of Done above contains the column road number, operation code, extra care, vehicle restricted and number of runs in each segment to complete the work. The column is created by extracting the data from the DigiRoad which shows the identification number of road segment. The operation code is taken from operation table. The operation will specify the requirements and requirements are taken from DigiContract. These requirements include the extra care column and vehicle restricted columns. Different digitalized values can be seen in the tables for these two columns. These values have different meanings. For example, the extra care 0 means that this segment does not require any special care. The extra care 1 means that this segment requires sand after

deicing. The extra care 2 means that this segment contains dangerous hill. Similarly, the vehicle restricted also contains different values. The value 0 for vehicle restricted means that every machine type is allowed to work on this segment whereas, value 1 would mean that tractors are not allowed on this segment. The number of runs is also shown by digitized values. The number of runs is calculated for both single carriageway roads and double carriageway roads. The number of runs is depicted in the Figure 34.

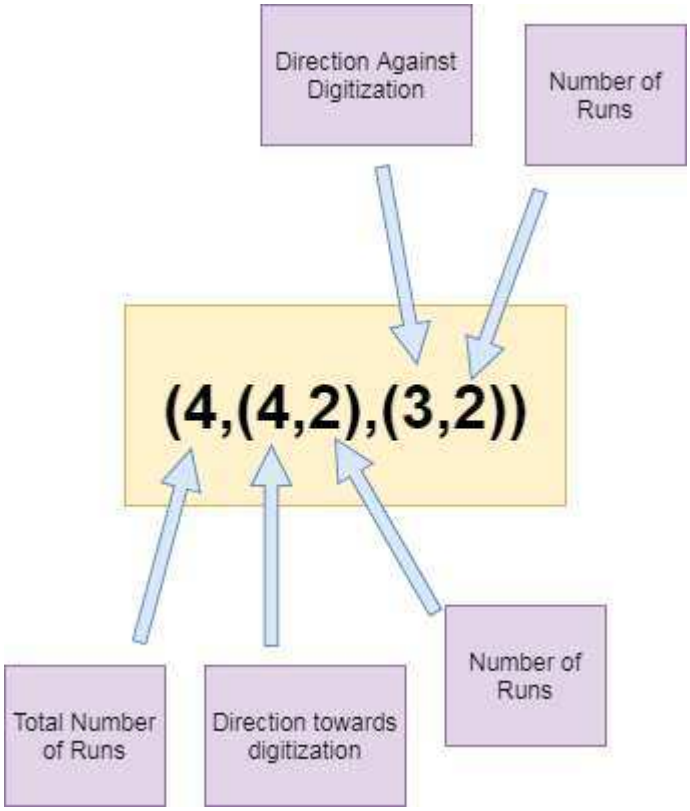


Figure 34: Number of runs in Definition of Done table

The first value after bracket shows the total number of runs required to complete the segment. The values in the next bracket show the direction of run and number of run to complete the segment. The direction is added in a similar way of DigiRoad. The direction of digitization is represented by a digitized value of 4 whereas direction against digitization is represented by number 3. In the Figure 34, it can be seen that 2 runs are required in both directions which are represented by 3 and 4. Similarly, if a bus stop is present on one of the direction this would mean that an extra run is required in that direction. These all digitized values formulate the table for Definition of Done.

6. CONCLUSION

This chapter concludes the overall thesis from the initial planning and problem identification phase to the results phase. Moreover, it also describes the need of future work that can be done in this research area.

6.1 Summary

Road maintenance is done as contracts. The service is provided by the contractors. The contractors provide service for an ordering authority. The contractor are forced to spend a lot of time in estimating the resources and tracking work progress because the contract data is not in computer processable form. However, this thesis has provided the novel approach to enable automated estimation and tracking of work progress by converting the contract data in computer processable form.

The technique of digitalization is used to convert the data from the human readable format to a form that computer can process. It converts the contract data in the digitalized form and allows the automation. The contract data contains the area map and requirements of the maintenance. To implement the proof of concept a real contract data has been taken and converted into digitalized form.

To convert data in the digitalized form different approaches have been applied. However, the best approach is implemented out of all selected approaches. In addition, to implement the best approach an exemplary contract data has been taken. First, the contract area map in contract data is represented as spatial data and second, the geometry of map is selected from data source named as DigiRoad. The geometry is selected from DigiRoad because it contains the road features. The road features are required because the contract area requirements are closely related to the features which will affect the road maintenance. The contract requirements are also converted into the digitalized form. The machinery information is also extracted and converted into digitalized form which is also needed to complete the contract area requirements. To enable automation, Definition of Done table is formulated using the data fusion technique. By fusing the machine data, digitalized contract requirements and DigiRoad geometry, different formulas are created which will evaluate the work done in an area according to the work requirements.

6.2 Future work

This thesis has presented a technique to allow automation but it has also opened the way to future work. There are different requirements in contract data that requires the weather information system. For example, there is a requirement on some roads that snow cleaning should start after 1 hour when snowing has stopped. By using the weather information system, the contractors will know in advance that when snow will be stop falling and they will keep resources ready.

The thesis has created the model that evaluates the work done in area. Similarly, a tracking model can be created. The tracking model will contain the real time tracking data of the working machines. The tracking model is then compared with the working model created in this thesis to show the status of work done in real time. Moreover, this can be later visualized to create a web application for showing the status of done in real time.

The research done has created a table for Definition of Done to evaluate work done according to requirements. However, if Definition of Done contains error, the system should detect it automatically. This can be achieved by using the techniques of machine learning. The techniques of machine learning will learn automatically by using the previous data and will correct errors in Definition of Done table.

In short the research work in this thesis has provided the techniques that can be utilized in the future to create a complete application that will show the status of done in a contract area for managers, street users and maintenance machine driver.

REFERENCES

[1] Pukhlov, I. (2014). Road maintenance in Russia and Finland, Saimaa University of Applied Sciences Technology Lappeenranta.

[2] Kubota, S., & Mikami, I. (2010). 4D information management system for road maintenance using GIS Satoshi Kubota Ichizou Mikami. *Proceedings of the International Conference on Computing in Civil and Building Engineering*.

[3] Bianchi, C. (2013). Implementation of road operation maintenance aspects in the planning and design phase, Chalmers University of Technology, Goteborg, Sweden.

[4] Chursin, G. (2015). Useful spatial data and GIS applications on the Internet for transportation companies, JAMK University of Applied Sciences.

[5] Mennecke, B. E., & Crossland, M. D. (1996). Geographic information systems: Applications and research opportunities for information systems researchers. System Sciences, 1996. *Proceedings of the Twenty-Ninth Hawaii International Conference On*, 3, 537–546.

[6] Koroleva, E. V., & Nikitin, Y. Y. (2014). U-max-statistics and limit theorems for perimeters and areas of random polygons. *Journal of Multivariate Analysis, 127, 98–111. https://doi.org/10.1016/j.jmva.2014.02.006*

[7] Memecke, B.E., Dangeimond, J., Santoro, P. J., Darling, M., & Crossland, M.D., (1995). Using geographic information systems as a tool for sensing and responding to customers. *In S.P. Bradley & R.L. Nolan (eds.), Multimedia and the Boundaryless World*, 1995 Harvard Business School Colloquium.

[8] Wei, W. (2012). Research on the Application of Geographic Information System in Tourism Management. *Procedia Environmental Sciences*, *12*(Icese 2011), 1104–1109. https://doi.org/10.1016/j.proenv.2012.01.394

[9] Stripple, H. (2001). Life cycle assessment of Road. *Swedish Environmental Research Institute IVL*, (March), 96p. https://doi.org/10.1177/0734242X10379146

[10] Erskine, M., Gregg, D., Karimi, J., & Scott, J. (2013). Business Decision-Making Using Geospatial Data: A Research Framework and Literature Review. *Axioms*, *3*(1), 10–30. https://doi.org/10.3390/axioms3010010

[11] Samet, H. (2009). Spatial Data Structures. Computer Science Department and Institute of Advanced Computer Studies and Center for Automation Research. University of Maryland.

[12] Pillai, M., & Adavi, P. (2013). Intelligent Contract Management. *International Journal of Scientific and Research Publications, Volume 3.*

[13] Watson, T. J. (2006). An Enterprise Electronic Contract Management System using Dual XML and Secure PDF Documents Thomas Kwok and Thao Nguyen IBM Research Division. *Enterprise Distributed Object Computing Conference Workshops 2006. 10th IEEE International.*

[14] Heikkila, R. (2013). Development of BIM based rehabilitation and maintenance process for roads, University of Oulu.

[15] Doktor-ingenieur, G. (2006). Integration of Spatial Vector Data in Enterprise Relational Database Environments Dissertation, University of Jena.

[16] Taha, A. (2005). Knowledge Discovery in GIS Data. Faculty of Computers & Information .Department of Information System, Cairo University

[17] Anon, 2017. Principles of Geographic Information Systems. *International Institute for Geo-Information Science and Earth Observation* (ITC).

[18] Esri, A., & Paper, W. (1998). ESRI Shapefile Technical Description. *Computational Statistics*, *16*(July), 370–371. https://doi.org/10.1016/0167-9473(93)90138-J

[19] Wang, L. (2015). Research Article A High Efficient Method of GIS Spatial Data Conversion. *Scholars Journal of Engineering and Technology (SJET).*

[20] R.J. Hijmans L. Guarino, M. C., & Rojas, E. (2001). Computer Tools for Spatial Analysis of Plant Genetic Resources Data: DIVA-GIS. *Plant Genetic Resources Newsletter*, (127), 15–19.

[21] Miler, M., Medak, D., & Odobasic, D. (2011). Two-Tier Architecture for Web Mapping with NoSQL Database CouchDB. *GI_Forum*, 1–10.

[22] Ayer, J., & Fosu, C. (2008). Map Coordinate Referencing and the use of GPS Datasets in Ghana. *Journal of Science and Technology (Ghana)*, *28*(1), 116–127. https://doi.org/10.4314/just.v28i1.33084

[23] Zlatanova, S. (2006). 3D Geometries in Spatial DBMS. *Innovations in 3D Geo Information Systems*, 1–14. https://doi.org/10.1007/978-3-540-36998-1_1

[24] Guting, R. H. (1994). An Introduction to Spatial Database Systems. *VLDB Journal, Special Issue on Spatial Database Systems*, *3*(4), 357–399.

[25] Shekar, S., & Chawla, S. (2003). Spatial Databases: A Tour. *Spatial Databases: A Tour*, 21–44. https://doi.org/papers3://publication/uuid/E626604D-1F38-4DCD-878A-C1E029178A87

[26] Toups, M. A. (2016). A study of three paradigms for storing geospatial data: distributed-cloud model, relational database, and indexed flat file, University of New Orleans.

[27] van Oosterom, P. J. M., Gorte, B. G. H., & Geomatics, M. (2012). Point Clouds in a Database. SPAR Europe 2014. Retrieved from http://repository.tudelft.nl/assets/uuid:58266d2c-330c-4493-a03a-d8b180eb810d/Master_thesis_-_report.pdf

[28] Batarseh, Feras A., and Avelino J. Gonzalez. (2009. Introduction to Data Analysis Handbook, 1st ed. Migrant & Seasonal Head Start Technical Assistance Center.

[29] Vosloo, J. J. (2014). A sport management program for educator training in accordance with the diverse needs of South African schools. Unpublished doctoral dissertation. North-West University, Potchefstroom.

[30] Rigaux, P., Scholl, M., & Voisard, A. E. (2002). Spatial Databases with Application to GIS, 2nd Ed. The Morgan Kaufmann Publisher Inc.

[31] Yeung, A. K. W., & Hall, G. B. (2007). Spatial Database Systems. Design, Implementation and Project Management, 2nd Ed. Springer-Verlog Newyork. https://doi.org/10.1007/1-4020-5392-4

[32] Ooi, B. C., Sacks-davis, R., & Han, J. (1995). Indexing in Spatial Databases, National University of Singapore *Most*, *2420*, 292–306. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.66.9499&rep=rep1&type=pdf

[33] SpatiaLite Cookbook. 2017. *SpatiaLite Cookbook*. [ONLINE] Available at: http://www.gaia-gis.it/spatialite-2.4.0-4/spatialite-cookbook/html/tech-intro.html. [Accessed 30 October 2017]

[34] Cuesta, H. (2010). Practical Data Analysis : An Example. Practical Data Analysis for Designed Experiments. https://doi.org/10.1007/978-1-84882-260-3

[35] Barker, K. N. (1980). Data collection techniques: observation. *American Journal of Hospital Pharmacy*, *37*(September), 1235–1243. https://doi.org/10.1097/00004583-200108000-00020

[36] DU, HESHAN, 2015. Matching disparate geospatial datasets and validating matches using spatial logic. University of Nottingham, 1-211.

[37] SCHA¨ FERS Michael., W. LIPECK Udo (2014). Spatial Data Integration Using Similarity-based Matching. Database Systems Group, Leibniz University of Hanover, Germany,

[38] Zhang, M., & Meng, L. (2007). An iterative road-matching approach for the integration of postal data. *Computers, Environment and Urban Systems*, *31*(5), 597–615. https://doi.org/10.1016/j.compenvurbsys.2007.08.008

[39] Christen, P. (2012). Concepts and Techniques for Record Linkage, Entity Resolution and Duplicate Detection, 3rd ed. Springer-Verlog Newyork. ISBN 978-3-642-31164-2

[40] Steffen Volz. (2006). An Iterative Approach for Matching Multiple Representations of Street Data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*(Part 2/W40), 101–110.

[41] Liu, C., Xiong, L., Hu, X., & Shan, J. (2015). A Progressive Buffering Method for Road Map Update Using OpenStreetMap Data. *ISPRS International Journal of Geo-Information*, *4*(3), 1246–1264. https://doi.org/10.3390/ijgi4031246

[42] Barbara, Santa. (2011). Automatically and Accurately matching objects in Geospatial datasets. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 38, Part II*.

[43] Beeri, C., Doytsher, Y., Kanza, Y., Safra, E., & Sagiv, Y. (2005). Finding corresponding objects when integrating several geo-spatial datasets. *Proceedings of the 2005 International Workshop on Geographic Information Systems - GIS '05*, 87. https://doi.org/10.1145/1097064.1097078

[44] Kraft, W. (1995). Entwurf von Zuordnungsalgorithmen zurnFortführung und Überprüfung von raumbezogenen Datenbeständen. Diploma Thesis at the Institute for Photogrammetry, University of Stuttgart, 75 pages.

[45] Filin, S., Doytsher, Y. (2000). Detection of Corresponding Objects in Linear-Based Map Conflation, Surveying and Land Information Systems. *International Journal of Geographical Information Science*.

[46] Rahm, E., & Do, H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, *23*(4), 3–13. https://doi.org/10.1145/1317331.1317341

[47] Van Den Broeck, J., Cunningham, S. A., Eeckels, R., & Herbst, K. (2005). Data cleaning: Detecting, diagnosing, and editing data abnormalities. *PLoS Medicine*, *2*(10), 0966–0970. https://doi.org/10.1371/journal.pmed.0020267

[48] Li, L. (2012). Data quality and data cleaning in database applications, University of Leipzig, Germany.

[49] Hellerstein, J. M. (2008). Quantitative Data Cleaning for Large Databases. *United Nations Economic Commission for Europe*, 42. Retrieved from http://db.cs.berkeley.edu/jmh/cleaning-unece.pdf%5Cnpapers2://publication/uuid/DC7173AB-6B26-4B8B-AEC3-4C7E65CEEFED

[50] Krishnan, S., Wang, J., Franklin, M. J., Goldberg, K., Kraska, T., Milo, T., & Wu, E. (2015). SampleClean: Fast and Reliable Analytics on Dirty Data. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 59–75. Retrieved from http://sites.computer.org/debull/A15sept/p59.pdf

[51] Müller, H., & Freytag, J. (2003). Problems, Methods, and Challenges in Comprehensive Data Cleansing. *Challenges*, (HUB-IB-164), 1–23. Retrieved from http://www.dbis.informatik.hu-berlin.de/fileadmin/research/papers/techreports/2003-hub_ib_164-mueller.pdf

[52] Raju, P. L. N. (2012). Spatial Data Analysis .*Satellite Remote Sensing and GIS Applications in Agricultural Meteorology*(2003)

[53] M., M. (2011). Spatial Data Analysis Model, Methods and Techniques, 4th ed. Heidelberg. Springer Science & Business Media.

[54] Fotheringham S, Rogerson P. (1994). Spatial Analysis and GIS, 1st ed. CRC Press

[55] Pfeiffer, D. U. (1996). Issues related to handling of spatial data. Australian Veterinary Association Second Pan Pacific Veterinary Conference, Christchurch, 23-28 June.

[56] Chiang, Y.-Y., Wu, B., Anand, A., Akade, K., & Knoblock, C. A. (2014). A system for efficient cleaning and transformation of geospatial data attributes. *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL '14*, 577–580. https://doi.org/10.1145/2666310.2666373

[57] Zhang, S., Yang, Q., Zhang, C., & City, M. (2002). Proceedings of the First International Workshop on Data Cleaning and Preprocessing. *Held in conjunction with ICDM 2002 Maebashi TERRSA, Maebashi City, Japan*

[58] Liggins, M., 2017. Handbook of Multisensor Data Fusion, 2nd ed. CRC Press.

[59] Wald, L., 2017. Data Fusion, 3rd ed. Presses des MINES

[60] Gros, X. E. (1997). Data Fusion - A Review. *NDT Data Fusion*, 5–42. https://doi.org/http://dx.doi.org/10.1016/B978-034067648-6/50004-9

[61] Bleiholder, J., & Naumann, F. (2008). Data fusion. *ACM Computing Surveys*, *41*(1), 1–41. https://doi.org/10.1145/1456650.1456651

[62] Pavlidis, P., Weston, J., Cai, J., & Noble, W. S. (2002). Learning Gene Functional Classifications from Multiple Data Types. *Journal of Computational Biology*, *9*(2), 401–411. https://doi.org/10.1089/10665270252935539

[63] R., J., 2015. Data Fusion Mathematics: Theory and Practice 4th ed . CRC Press.

[64] Castanedo, F. (2013). A review of data fusion techniques. *ScientificWorldJournal*, *2013*, 704504. https://doi.org/10.1155/2013/704504

[65] Abdelgawad, A., & Bayoumi, M. (2012). Resource-Aware Data Fusion Algorithms for Wireless Sensor Networks. Springer Science & Business Media , *118*. https://doi.org/10.1007/978-1-4614-1350-9

[66] Digiroad documents - Finnish Transport Agency. 2017. Digiroad documents - Finnish Transport Agency. [ONLINE] Available at: https://www.liikennevirasto.fi/web/en/open-data/digiroad/documents#.WgiGnluCzIV. [Accessed 12 November 2017].

[67] GeoJSON. 2017. GeoJSON. [ONLINE] Available at: http://geojson.org/. [Accessed 30 November 2017].

[68] Wiley, B. (2009). GPS Geodetic Reference System WGS-84. *International Committee on GNSS*, (September).

[69] European Terrestrial Reference System 89 (ETRS89) - Knowledge Base English - QPS Confluence. 2017. European Terrestrial Reference System 89 (ETRS89) - Knowledge Base English - QPS Confluence. [ONLINE] Available at: https://confluence.qps.nl/pages/viewpage.action?pageId=29858197. [Accessed 30 November 2017].

[70] PostgreSQL: The world's most advanced open source database. 2017. PostgreSQL: The world's most advanced open source database. [ONLINE] Available at: https://www.postgresql.org/. [Accessed 30 November 2017].

[71] Li, D., Wang, S., & Li, D. (2015). Concepts, Principles and Applications of Spatial Data Mining and Knowledge Discovery. *ISSTM 2005*. https://doi.org/10.1007/978-3-662-48538-5

[72] Welcome to the QGIS project!. (2017). Welcome to the QGIS project!. [ONLINE] Available at: https://www.qgis.org/en/site/. [Accessed 10 December 2017].